InferenceMAP: Mapping of Single-Molecule Dynamics with Bayesian Inference

Mohamed El Beheiry[1,2], Maxime Dahan[1] and Jean-Baptiste Masson[2,3]

[1] Physico-Chimie Curie, Institut Curie, CNRS UMR 168, Université Pierre et Marie Curie-Paris 6, 26 rue d'Ulm 75248 Paris, France
[2] Institut Pasteur, Physics of Biological Systems, CNRS UMR 3525, 25/28 rue du Docteur Roux, 75015 Paris, France
[3] Janelia Research Campus, Howard Hughes Medical Institute, Ashburn, VA 20147, USA

Single-particle tracking (SPT) grants unprecedented insight into cellular function at the molecular scale [1]. Throughout the cell, the movement of single-molecules is generally heterogeneous and complex. Hence, there is an imperative to understand the multi-scale nature of single-molecule dynamics in biological systems. We have previously shown that with high-density SPT, spatial maps of the parameters that dictate molecule motion can be generated to intricately describe cellular environments [2,3,4]. To date, however, there exist no publically available tools that reconcile trajectory data to generate the aforementioned maps. We address this void in the SPT community with InferenceMAP: an interactive software package that uses a powerful Bayesian method to map the dynamic cellular space experienced by individual biomolecules (**Supplementary Software**).

High-density SPT methods, such as sptPALM [5] and uPAINT [6], capture thousands of molecule trajectories in a few minutes of acquisition at high spatiotemporal resolution. By using one of the many available particle-tracking algorithms [7], trajectories are reconstituted and accepted as input to InferenceMAP. This massive trajectory data is thereafter treated with a Bayesian inference algorithm that, notably, imposes no constraints on trajectory lengths [2,3,4]. Our algorithm is compatible with different models of single-molecule motion, including hopping diffusion (**Fig. 1a**), active processes, confinement, and interaction energy-driven systems (**Fig. 1b**). Model-specific physical processes are distinguished and mapped, revealing rich landscapes of molecule dynamics.

Generating dynamical maps from single-particle trajectories is critically dependent on the meshing utilized. As local diffusivities may vary by orders of magnitude in a few hundred nanometers, meshes should locally adapt to match the characteristic size of molecule displacements. To this end, InferenceMAP offers adaptive meshing techniques that users may tune to fit the spatial organization of their single-molecule trajectories (**Fig. 1a,c**). In each zone of a mesh, dynamic parameters are inferred to give rise to a parameter landscape (**Supplementary Information**).

Calculations can be performed in an automated fashion irrespective of the biological system, or parameters may be carefully adjusted to conform to desired mapping resolution and optimization constraints. Additionally, beforehand knowledge of the biological system can be incorporated in calculations via user-defined prior probabilities (**Supplementary Information**). Furthermore, a randomized optimization algorithm is available for exceptionally large problems, permitting mapping of entire cells such as in **Fig. 1b**. Highly-localized analysis in sub-regions is easily performed through a custom-selection macro. Resulting inferred data is exportable in image and ASCII formats.

InferenceMAP offers a host of features to address a major concern of the single-particle tracking community: it reveals the parameters that dictate the motion of molecules. Moreover, in the burgeoning age of big data experimental biophysics, it is an important contribution that extracts sensible results from otherwise dense and complicated observations using a robust Bayesian method. InferenceMAP is controlled with a user-friendly interface and is compatible with Mac OS X and Windows. The software is freely available for academic use (source code is available upon signature of a Material Transfer Agreement), and latest versions may be downloaded from http://umr168.curie.fr/en/research-groups/locco/software and from http://www.pasteur.fr/en/research/genomes-genetics/units-groups/jean-baptiste-masson. The authors request acknowledgment of the use of InferenceMAP in published works.

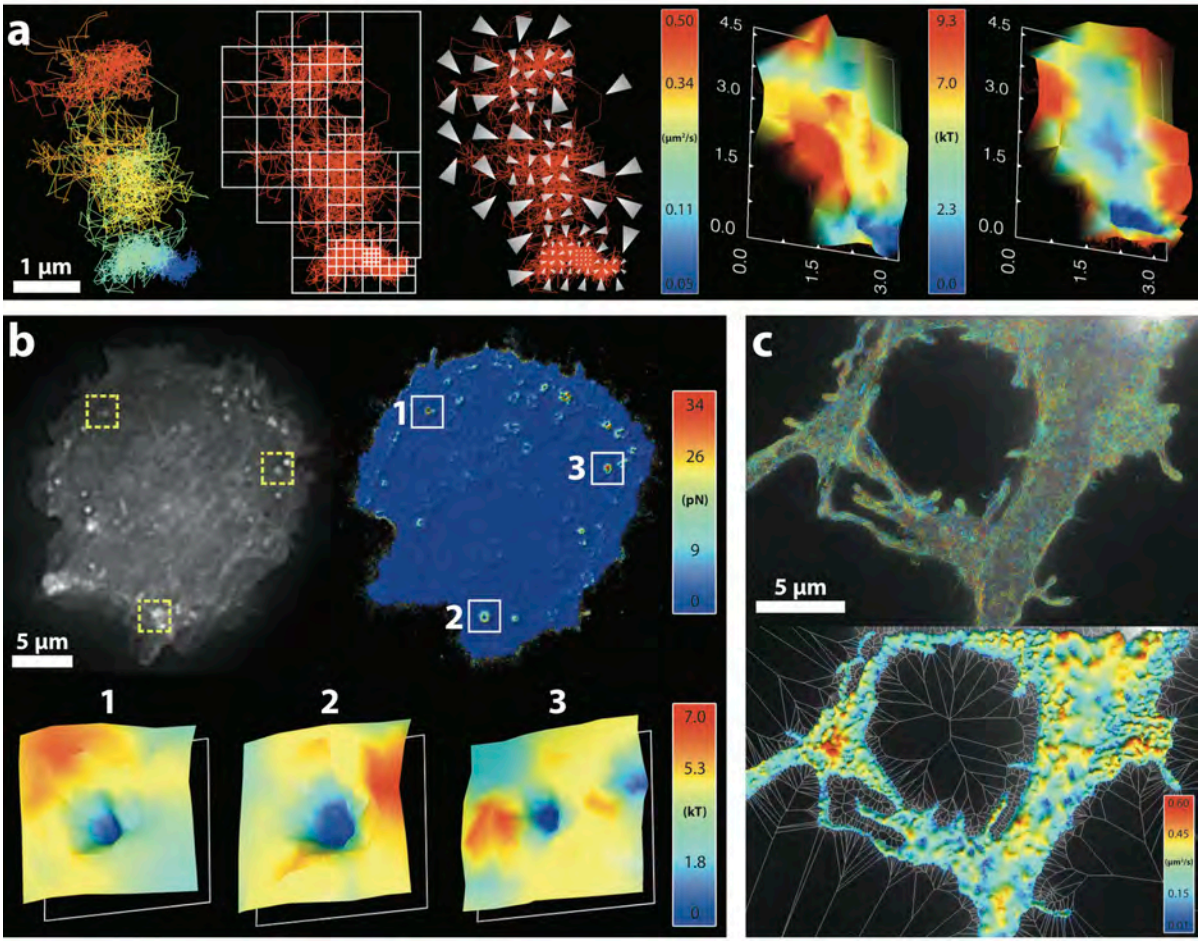


Figure 1 Caption

Fig 1. Various uses of InferenceMAP on experimental single-particle trajectories. Shown are various ways to meshing the area explored by trajectories and to represent inferred dynamical parameters (diffusivity, force components, and interaction potentials).

a. Typical analysis workflow in InferenceMAP, from left to right. First panel: trajectory of an ε -toxin receptor tagged to an amine-coated lanthanide oxide nanoparticle on an MDCK cell hopping between different lipid rafts [3]. The color code is associated to time, from blue (beginning) to red (ending) for a duration of 290 s. Second panel: same trajectory in red, overlaid with a quad-tree mesh. Note the multi-scale nature of the mesh, which depends on both the number of points and the local diffusivity. Third panel: force map (directional bias) acting on the receptor in its various confinement areas, indicated by white arrows. Fourth panel: Three-dimensional landscape view of diffusivity. Fifth panel: Landscape view of interaction energy map, revealing three membrane confinement domains.

b. Diffusivity map of glycine receptor construct (pHluorin-TMD-β-Loop 400AA [4]) in a mouse hippocampal neuron from a uPAINT measurement with an anti-GFP antibody coupled to atto647N. Top panel: 9,453 trajectories of the GlyR construct overlaid to an ensemble (GFP) fluorescence image of receptors. Color code distinguishes different trajectories. Bottom panel: three-dimensional diffusivity map plotted overtop the employed Voronoi tessellated mesh. The Voronoi mesh allows for efficient mapping of cells with complex geometries, and clearly shows local variability of the diffusivity.

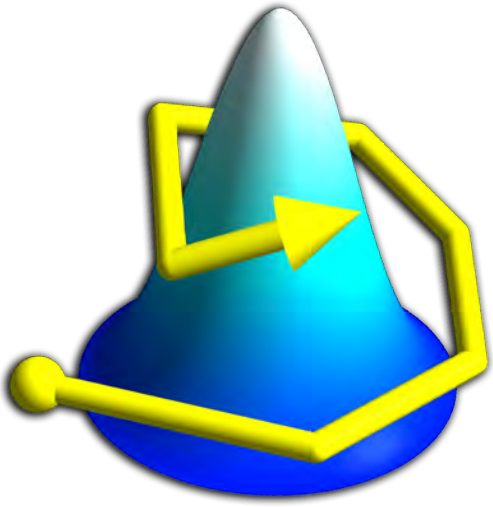
c. Whole-cell interaction energy map of glycine receptor construct (Dendra2-TMD-ß-Loop WT) with gephyrin clusters using TIRF microscopy. Top-left panel: ensemble fluorescence image of Cerulean-Gephyrin clusters (yellow boxes) in a COS-7 cell. Top-right panel: Three-dimensional map of force amplitudes at the basal membrane of the cell. Note that the map is mostly flat, except in highly-localized regions colocalized with gephyrin clusters (white boxes). Interaction energy wells corresponding to the boxed regions are indicated in the bottom three panels.

Acknowledgments

We thank A. Alexandrou, D. Casanova, S. Türkcan, and M. Richly for providing lipid raft data. For glycine receptor constructs, gephyrin plasmids, and neuronal datasets, we thank C. Salvatico, P. Dionne, C. Specht, M. Renner, and A. Triller. We also thank C. Zimmer, J.-C. Olivo-Marin, and D. Krapf for useful discussions in the preparation of this work. This work was supported by funding from the state program "Investissements d'avenir," managed by Agence Nationale de la Recherche (Grant ANR-10-BINF-05 "Pherotaxis" and Grant ANR-10-INSB-04 'France BioImaging »), the Institut Curie International PhD Program, Paris-Science-Lettres (program ANR-10-IDEX-0001-02 PSL), and from ANR grants TRIDIMIC and SYNAPTUNE.

References

1. Kusumi, A. *et al*. *Nat. Chem. Biol.* **10**, 524–532 (2014).
2. Masson, J.-B. *et al. Phys. Rev. Lett* **102**, 048103 (2009).
3. Türkcan, S. *et al. Biophys. J.* **102**, 2299–2308 (2012).
4. Masson, J.-B., *et al*. *Biophys. J.***106** 78–83 (2014).
5. Manley, S., *et al. Nat. Methods* **5**, 155–157 (2008).
6. G. Giannone, G. *et al. Biophys. J.* **99**, 1303–1310 (2010).
7. N. Chenouard, *et al. Nat. Methods* **11**, 281–289 (2014).



InferenceMAP User Manual

Version 1.0

Mohamed El Beheiry
Institut Curie - Centre de Recherche
Laboratoire Physico-Chimie Curie (UMR168)
26, rue d'Ulm
75248 Paris, France
mohamed.elbeheiry@gmail.com

Jean-Baptiste Masson
Institut Pasteur
Physics of Biological Systems
25/28 rue du Docteur Roux
75015 Paris , France
jbmasson@pasteur.fr

InferenceMAP is registered with the *Agency for the Protection of Programs* (APP) under reference **FR.001.350042.000.S.P.2014.000.20700**

Contents

1 Preliminary Remarks **3**
1.1 Models of Motion . . . 3
1.2 Constraints . . . 3
1.3 Tracking Software . . . 4

2 Installation & Execution **5**
2.1 Windows (XP & 7) . . . 5
2.1.1 Known Issues . . . 5
2.2 Mac OS X . . . 5
2.2.1 Known Issues . . . 5

3 File Formats **6**
3.1 x y t File (.xyt) . . . 6
3.2 tr x y t File (.trxyt) . . . 6

4 Main Interface **7**

5 Trajectory Visualization **8**

6 Meshing **9**
6.1 Square Meshing . . . 9
6.2 Voronoi Tessellation . . . 10
6.2.1 Clustering . . . 10
6.2.2 Tessellation . . . 11
6.3 Quad-Tree Meshing . . . 11
6.4 Meshing Advice . . . 12
6.4.1 Neighboring Zone Connections . . . 12
6.4.2 Manual Zone Activation and Deactivation . . . 12
6.4.3 Preventing Holes in the Mesh . . . 13

7 Inference **14**
7.1 Inference Workflow . . . 14
7.2 (D) Inference . . . 15
7.2.1 Applicability . . . 16
7.3 (D,F) Inference . . . 16
7.3.1 Potential Calculation . . . 16
7.3.2 Applicability . . . 16
7.4 (D,Drift) Inference . . . 17
7.4.1 Applicability . . . 17
7.5 (D,V) Inference . . . 17
7.5.1 Applicability . . . 18
7.6 Polynomial Potential Inference . . . 18
7.6.1 Applicability . . . 18
7.7 Randomized Optimization . . . 18
7.8 Freehand Selection Inference . . . 19

8 Priors **21**
8.1 Uniform . . . 21
8.2 Jeffreys . . . 21
8.3 Smoothing . . . 21

9 Posterior Sampling **23**

10 Trajectory Simulation **25**

11 Landscape Viewing **26**

12 Freehand Selection **27**

13 Tools **28**
13.1 Annotations . . . 28
13.2 Density Calculation . . . 28
13.3 Interval Selection . . . 29
13.4 Save Screen . . . 29
13.5 TIFF Overlay . . . 29
13.6 White Background . . . 30

14 Performance **31**
14.1 Trajectory Length . . . 31
14.2 Number of Localizations . . . 31
14.3 Mode Selection . . . 31
14.4 Prior Probabilities . . . 33

15 Stepped-Through Examples **34**
15.1 Example 1: Generating a Diffusion Map . . . 34
15.2 Example 2: Generating a Potential (Interaction) Energy Map . . . 37
15.3 Example 3: Membrane Microdomain . . . 42
15.4 Example 4: Neurotransmitter Receptors . . . 45

16 Acknowledgments **50**
16.1 Development Libraries . . . 50
16.2 FLTK . . . 50
16.3 OpenGL . . . 50
16.4 Libtiff . . . 50
16.5 TexFont . . . 50
16.6 Clustering Algorithms . . . 50

17 Contact Information **51**

1 Preliminary Remarks

The trajectory inference technique utilized by **InferenceMAP** (first described in [8]) is used to study the dynamics of single molecules. Specifically, it generates maps of the dynamic parameters that dictate the motion of single molecules. These parameters may include the diffusion, forces (directional biases), interaction (potential) energies, and drift. Mapping these spatially-dependent parameters is of great interest for a variety of studies. First, dynamical maps can be used to decipher physical interactions a molecule may have with its surroundings. Secondly, generated maps reliably distinguish physical parameters from one another. Finally, it is a means of probing the strength of single molecule interactions.

To date, this technique has been used to study the dynamics of single molecules in many biological contexts, these include:

- Neurotransmitter receptors [1]
- Membrane microdomains [6]
- Transcription factors
- Viral capsid fusing proteins
- Proteins on unilamellar vesicles

1.1 Models of Motion

The widespread applicability of **InferenceMAP** is largely due to the different models of single molecule motion it supports. Below the models are briefly described (a full description is available in Section 7).

- **Diffusion Only.** Solely the diffusivity is estimated from the trajectories.
- **Diffusion and Force.** Local force components in addition to the diffusion are inferred, with an option to estimate interaction energies.
- **Diffusion and Drift.** Local drift (speed) in addition to the diffusion are inferred. This model is applicable to systems possessing active processes, where forces may not be conservative.
- **Diffusion and Potential.** Potential (interaction) energy, force, and diffusion components are estimated from trajectories.
- **Polynomial Potential.** Potential (interaction) energy, force, and diffusion components are estimated from trajectories. This model is applicable to small regions where there is trajectory confinement.

1.2 Constraints

Users should be aware of constraints regarding the single molecule trajectory inference technique implemented in **InferenceMAP**. With its various inference modes and features, **InferenceMAP** handles trajectory motion that can be modeled by an *overdamped Langevin* equation. This model is a good approximation to memoryless (Markovian) motion, which single molecules typically exhibit.

Below, potential constraints to the use of **InferenceMAP** are listed.

- **Limited Localization Density**. If datasets have a low number of trajectory points (i.e. localizations), the precision of inferred parameters may be impeded. A possible way to get around this issue is to reduce the spatial resolution of the mesh in which parameters are inferred. General rules for localization numbers are given in Section 6.
- **Timescale of Dynamics**. Users should be aware of the general timescale of dynamics within their system (e.g. the duration of an interaction or a transport event). **InferenceMAP** allows trajectories to be temporally windowed to accommodate different event durations.
- **Viscoelastic Motion**. Viscoelastic motion is not accurately described by the overdamped Langevin equation, in which case **InferenceMAP** is not advised for estimating dynamics. Such motion is observed in the motion of large cytosolic vesicles, for example.
- **Immobile Trajectories**. The overdamped Langevin equation does not describe entirely immobilized trajectories. In cases where trajectories have mixed mobile and immobile populations, it is advised to segregate populations prior to analysis with **InferenceMAP** .

1.3 Tracking Software

As **InferenceMAP** is used downstream from the single molecule tracking step (it takes trajectories as input data), a tracking software is needed to reconstruct trajectory translocations from acquired microscopy images. A few of the freely available tracking software tools are listed in the references below.

- **u-track** Jaqaman et al., *Nature Methods* 5, pp. 695–702 (2008)
- **MTT** Sergé et al., *Nature Methods* 5, pp. 687–694 (2008)
- **Various** Chenouard et al., *Nature Methods* 11, pp. 281–289 (2014)

2 Installation & Execution

2.1 Windows (XP & 7)

InferenceMAP has been tested on the **Windows XP**, **Windows 7**, and **Windows 8** operating systems. The installation instructions are as follows:

1. Assure that the **Microsoft Visual C++ 2010 Redistributable Package** is installed. It can be found here: **http://www.microsoft.com/en-us/download/details.aspx?id=5555**.
2. Double-click the install package **InferenceMAP Windows.zip** and follow the on-screen installation instructions.
3. Double-click the **InferenceMAP.exe** in the install directory to start the program.

The execution of **InferenceMAP** is coupled with that of a Windows **Command Prompt**. The purpose of this is to display the progress in potentially lengthy calculations.

2.1.1 Known Issues

Reported issues with the **Windows** version of **InferenceMAP** may potentially be circumvented trying the following steps:

1. Place the contents of the ViSP installation package directly in your HD path (e.g. place them in a directory such as **C:/InferenceMAP/**).
2. Try running ViSP after renaming the .dll files in the installation directory (e.g. rename **opengl32.dll** to **opengl32_temp.dll** and see if it runs).
3. Assure that you have the latest drivers for your graphics card installed.

2.2 Mac OS X

InferenceMAP has been tested on versions **10.6.x** (Snow Leopard), **10.7.x** (Lion), **10.8.x** (Mountain Lion), and **10.9.x** (Mavericks) of the **Mac OS X** operating system. The installation instructions are as follows:

1. Mount **InferenceMAP for Mac.dmg** by double-clicking it.
2. Double click the **InferenceMAP.pkg** file and follow the on-screen installation instructions.
3. Click the **InferenceMAP** file in the chosen installation directory to start the program.

2.2.1 Known Issues

Reported issues with the **Mac** version of **InferenceMAP** may potentially be circumvented trying the following steps:

1. If opening **InferenceMAP.app** presents the message:
   "InferenceMAP" can't be opened because it is from an unidentified developer.
   Instead, right-click **InferenceMAP.app** and select **Open**. The following the message will appear:
   "InferenceMAP" is from an unidentified developer. Are you sure you want to open it?
   Select **Open** to launch **InferenceMAP** .
2. If opening **InferenceMAP.app** presents the message:
   "InferenceMAP.app" is damaged and can't be opened. You should move it to the Trash.
   Go to **System Preferences**, select **Security & Privacy**. Choose the **General** tab. Click the lock in the bottom-left corner, after which you will be request to enter your user credentials and password. Under the **Allow apps downloaded from:**, select **Anywhere**. Now, when launching **InferenceMAP.app** a window will appear showing **"InferenceMAP.app" is an application downloaded from the Internet. Are you sure you want to open it?**, select **Open**.

3 File Formats

InferenceMAP supports two ASCII (text) file formats for single-particle trajectory information:

1. Columns are tab-delimited
2. All geometric units are in micrometers [µm]
3. Temporal information is given in seconds [s]

Files are opened by selecting **Open** in the **File** menu. Up to 10 files may be opened in a session, tabs for which will appear in the **File Tabs** panel of the main interface (see Section 4).

3.1 x y t File (.xyt)

This format is for *single* trajectories. The user has the option of selecting multiple files of this type in the **File Open** dialog box (each one corresponding to a single trajectory). Columns are defined as follows:

Column 1	**Column 2**	**Column 3**
x-Coordinate [µm]	y-Coordinate [µm]	Time Stamp [s]

3.2 tr x y t File (.trxyt)

This format is for *multiple* trajectories. Columns are defined as follows:

Column 1	**Column 2**	**Column 3**	**Column 4**
Trajectory Number	x-Coordinate [µm]	y-Coordinate [µm]	Time Stamp [s]

4 Main Interface

InferenceMAP is controled with an interactive graphical user interface, as shown in Figure 1.

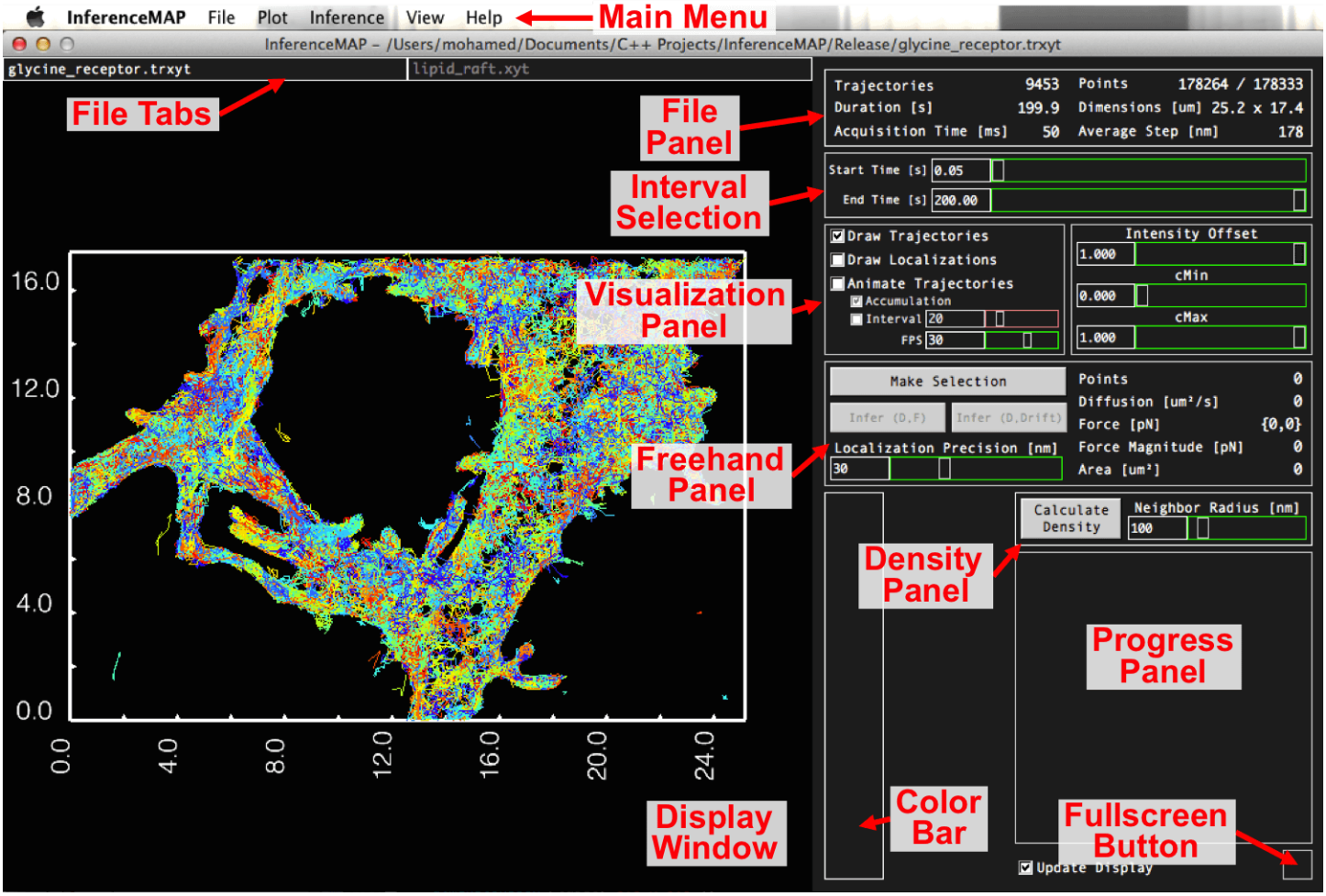


Figure 1: **InferenceMAP** main interface.

Main Menu – Access to input/output options and all the main functions in **InferenceMAP**

Display Window – Window displaying current trajectory data set

File Tabs – Indication of files opened in current session

File Panel – Spatiotemporal data concerning the current file

Interval Selection – Select to activate trajectories within a specified time window

Freehand Panel – Macro for selecting a custom region in the current file and performing a direct trajectory inference

Color Bar – Color code corresponding to the selected meshing parameter overlay

Visualization Panel – Current file viewing options

Density Panel – Calculate the localization density for the displayed trajectories

Progress Panel – Displays calculation progress

Fullscreen Button – Toggle between full and default screen size

5 Trajectory Visualization

Upon starting **InferenceMAP**, trajectories may be inspected with various visualization functions. After loading a trajectory file (via the **File** menu), data may be visualized with options in the **Visualization Panel**, shown in Figure 2.

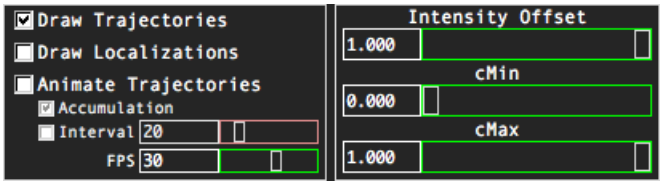
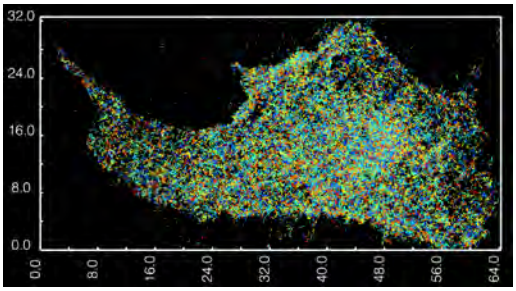


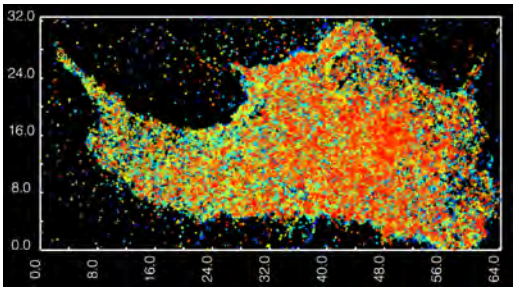
Figure 2: Visualization Panel in the main interface of **InferenceMAP** .

For a loaded trajectory, the user has access to the following visualization options:

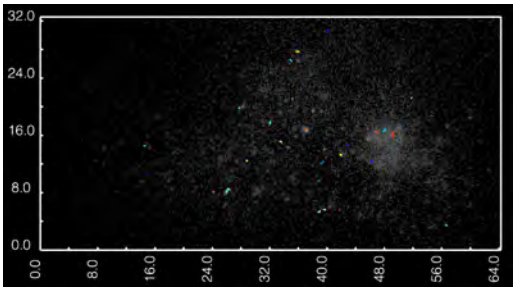
Draw Trajectories. Displays all trajectories in loaded file. For **.trxyt** files, trajectories are distinguished by different colors.



Draw Localizations. Draws localizations (points) from all trajectories (color code is corresponds to time of appearance).



Animate Trajectories. Animates the trajectories either in: **Accumulation** mode which animates and accumulates all the trajectories in the loaded file, or in an **Interval** mode which animates the trajectories in a "sliding window" mode, in which the interval size may be adjusted with its corresponding slider (20 steps by default). The **FPS** slider adjusts the frames-per-second of animations in the **Display Window**.



Trajectory viewing may additionally be synchronized to raw acquisition films, using the **TIFF Overlay** tool described in Section 13.5.

6 Meshing

InferenceMAP offers three types of meshing, shown in Figure 3. The type of meshing method deemed appropriate for a given data set depends principally on the density of localizations (not necessarily trajectories), and, not unrelated, the desired resolution with which the dynamic landscape is mapped. The user is reminded that the main assumption in meshing is that the inferred parameter (diffusion coefficient, force components, drift components, or potential energy) remains constant in each zone.

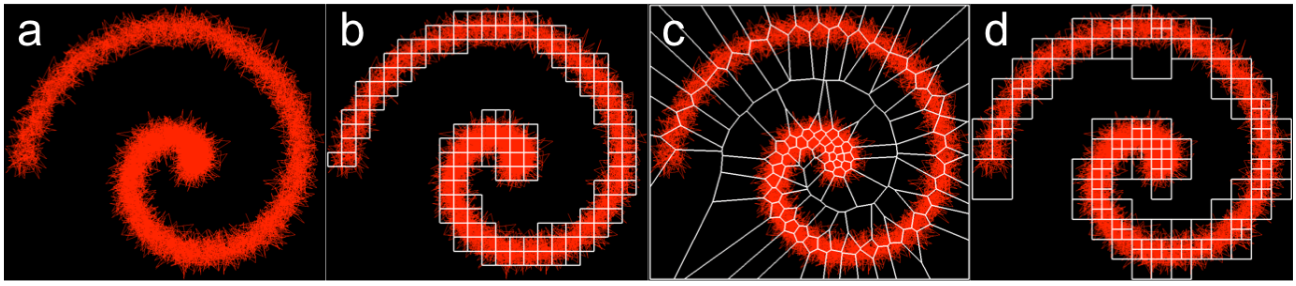


Figure 3: Meshing types available in **InferenceMAP** for a simulated spiral trajectory (a): (b) Square Meshing, (c) Voronoi Meshing, and (d) Quad-Tree Meshing

6.1 Square Meshing

Square Meshing involves spatial partitioning the trajectory overlay space into identically-sized squares. It is most appropriate in cases where the localization density throughout the data set is relatively homogeneous. The **Square Meshing** dialog box is shown in Figure 4.

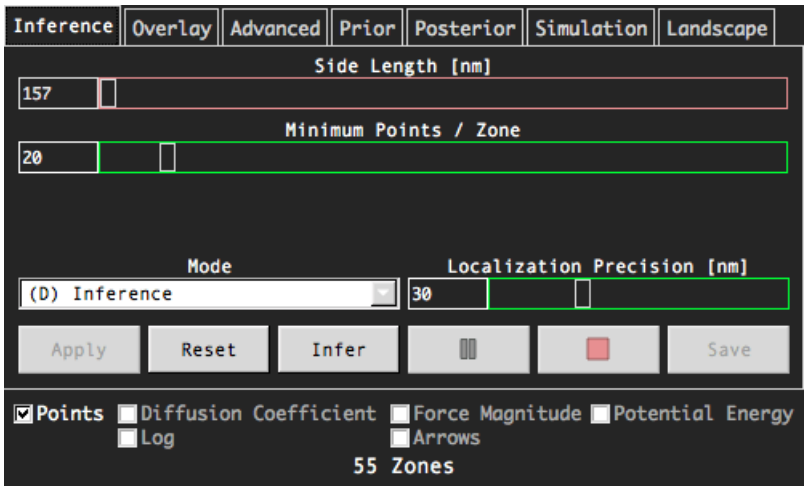


Figure 4: **Square Meshing** interface.

An appropriate option for the square side length is the average trajectory step length (this is the minimal and initial default value in the **Side Length** slider of the **Square Meshing** interface). However, if trajectory points (localizations) are not sufficiently dense, the side length should be increased (at least 20 points per zone is recommended for accurate parameter estimation).

To apply a square mesh to the current file, press the **Apply** button. This will create the internal data structure necessary to perform the inference calculation.

The **Minimum Points / Zone** slider permits filtering of mesh zones based on the number of localizations contained in them. It is generally recommended to have roughly ~ 20 points per cell. This notably adds constraints to the selected zone side length. In the case that the side length needs to be readjusted, the user may select the **Reset** button, which reinitializes

the mesh selection options.

The fields in the **Mode** drop-down menu are described in detail in Section 7.

6.2 Voronoi Tessellation

Voronoi Tessellation is one of the adaptive meshing methods available in **InferenceMAP**. It is most appropriate in cases where there is significant heterogeneity in the density of localizations (not necessarily trajectories). In contrast to **Square Meshing**, it will generate more zones in regions where localizations are more dense, and additionally adapt the size of zones based on the density of localizations within it. that eventually mesh generation includes two steps: **Clustering** and **Tessellation** which are discussed in Sections 6.2.1 and 6.2.2, respectively.

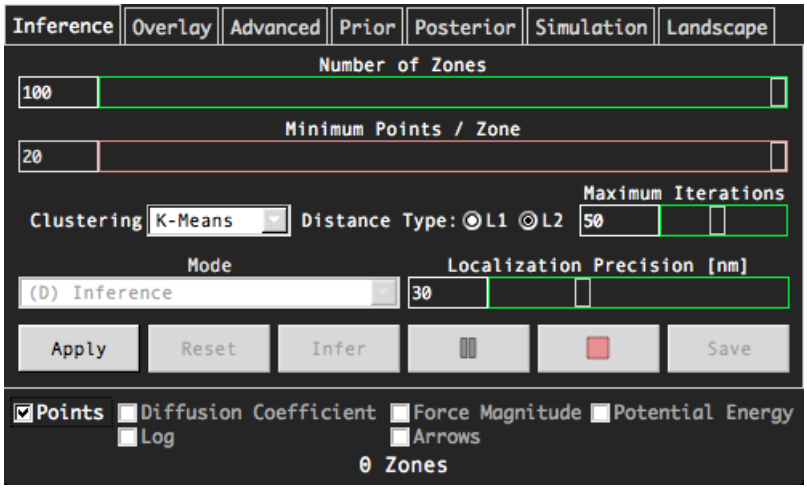


Figure 5: **Voronoi Tessellation** interface.

The Voronoi tessellated mesh is generated by pressing the **Apply** button. This will create the internal data structure necessary to perform the inference calculation.

6.2.1 Clustering

The first step to generate a Voronoi tessellated mesh is to cluster (or group) localizations together in a supervised fashion. The different modes of clustering are available in the **Clustering** drop-down menu in the **Voronoi Tessellation** interface (Figure 5). The available modes include:

- **K-Means** – Localizations are clustered to globally minimize the within-cluster sum of squares of all the clusters.
- **H-Means** – A simpler algorithm than K-Means, which assigns localizations to the closest randomly-chosen cluster centers. After assignment, clusters centers are replaced by the centroid, and the process is iteratively repeated.

Additionally, there is the option to choose between square-distance minimization (L2) or absolute distance minimization (L1), available in the **Voronoi Tessellation** interface, seen in Figure 5. Maximum clustering iterations can also be specified by the user in the **Maximum Iterations** slider.

The clustering algorithms used are based on those from John Burkardt, available at `http://people.sc.fsu.edu/~jburkardt/c_src/asa136/asa136.html`.

An additional reference for Voronoi Tessellation is *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams* by Atsuyuki Okabe, et al.

6.2.2 Tessellation

The number of clusters (which will correspond to the number of cells in the generated mesh) is specified with the **Number of Zones** slider in the **Voronoi Tessellation** interface, seen in Figure 5. Based on the clusters defined from Section 6.2.1, a Voronoi diagram is generated. Effectively, it involves spatially partitioning clusters into convex polygons, such that each of the clustered points inside the polygon is closest to its associated barycenter than to any other.

An important point to consider is that there is no restriction on the minimal dimensions of the Voronoi polygons, meaning the characteristic dimension of zones in the voronoi mesh may end up smaller than the average trajectory step-size. If this is observed, the simplest way to avoid the appearance of such zones is to regenerate the mesh with a smaller number of cells (specified in the **Number of Zones** slider). In practice, however, even with such zones present in the generated mesh, the final calculated parameter values are not greatly perturbed.

6.3 Quad-Tree Meshing

Quad-tree Meshing is an adaptive meshing method that recursively generates subzones (*leaves*) based on a localization *capacity* metric. Being a meshing technique that conforms to the density of localizations, it is especially relevant in cases where trajectory densities are strongly heterogeneous.

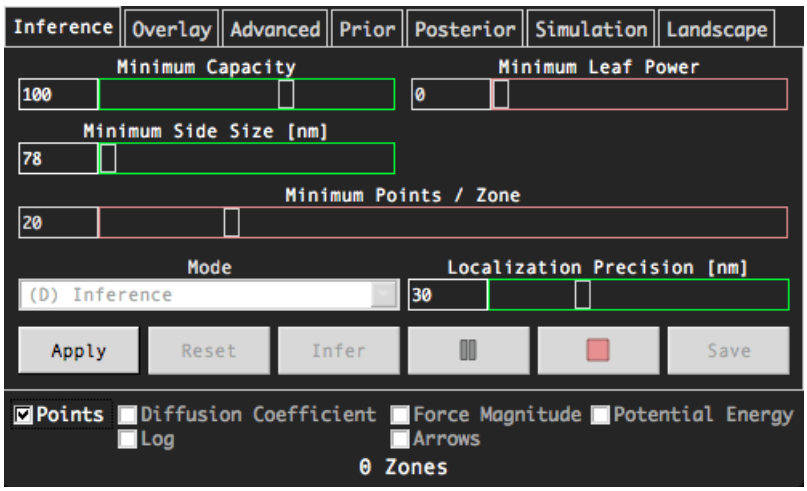


Figure 6: **Quad-Tree Meshing** interface.

Algorithmically, the quad-tree mesh is generated through the addition of localizations into a single square region. Points are added sequentially, until the *capacity* is exceeded. At this point, the mesh is subdivided into four identical squares. This process recursively takes place as more points are added, until no zones exceed the user-specified capacity, although some constraints may prevent this from being the case in practice (see below). The result is a hierarchical mesh.

A few constraints (related to trajectory overlay data) distinguish the quad-tree implementation in **InferenceMAP** to those that may be used for data structures and other applications:

- The side length of zones are limited (by default) to the average trajectory step size of the trajectories. This setting may be adjusted by selecting the minimum side length in the **Minimum Side Size** slider.
- There is an additional iteration at the end of the generation of the quad-tree mesh, which will act to ensure subdivided zones have above the minimal number of points required (by stepping up the quad-tree structure). This is why certain zones may have more than the capacity defined in the **Minimum Capacity** slider.
- Subzones may be filtered based on their *power* (side length) with the **Minimum Leaf Power** slider.

The quad-tree mesh is generated by pressing the **Apply** button. This will create the internal data structure necessary to perform the inference calculation.

6.4 Meshing Advice

In general, there are some key considerations the user should keep in mind when generating a mesh for the inference calculation described in Section 7. This section describes these considerations in some detail.

6.4.1 Neighboring Zone Connections

Calculation of the potential (interaction) energy in a given zone necessarily depends on the potential values in neighboring zones. **InferenceMAP** enables the user to define the neighbors of given zones via the **Advanced** tab of the meshing interface window, shown in Figure 7. Invalid neighbor connections can bias the inference calculation and may reduce the accuracy of the potential energy map.

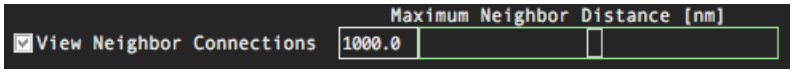


Figure 7: Neighbor connection options in **Advanced** tab of meshing interface.

In adjusting the **Maximum Neighbor Distance** slider, the user can remove connections between zones that are not physically related. These connections are indicated with yellow lines connected between the barycentres of the localizations in each mesh zone.

These erroneous connections can be particularly apparent in the adaptive meshing types (the **Quad-Tree** and **Voronoi Tessellation**). Figure 8 shows how erroneous connections may be removed between zones to ensure a more accurate potential calculation.

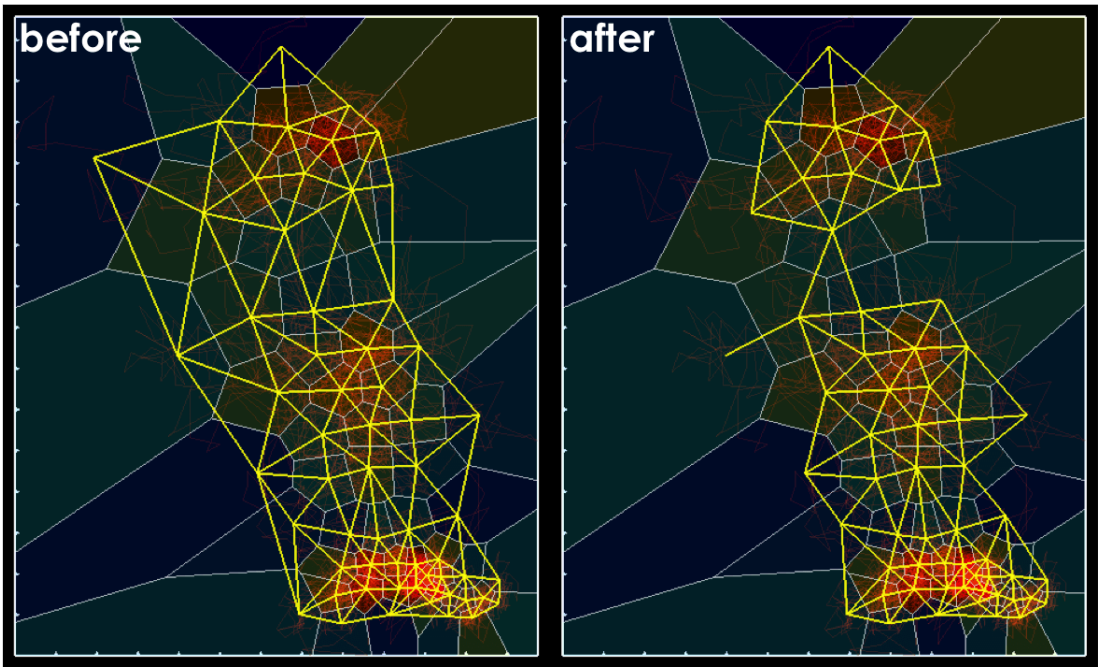


Figure 8: In adjusting the **Maximum Neighbor Distance** slider, connections between neighboring zones can be added or removed (indicated with yellow lines). This figure demonstrates the removal of erroneous neighboring zone connections.

6.4.2 Manual Zone Activation and Deactivation

After a mesh has been applied, prior to the inference calculation, the user may *manually* activate and deactivate zones. The motivation here is to add flexibility to the otherwise non-flexible zone filtering options (e.g. **Minimum Points**, **Minimum Leaf Power**, **Maximum Neighbor Distance**, etc.). Mesh zones are made active or inactive by selection or deselection via *right-clicking* with the mouse.

6.4.3 Preventing Holes in the Mesh

In general, "holes" (inactive zones surrounded by active ones) should be avoided. This may perturb the potential calculation, as it can greatly affect local values of the potential gradient that may propagate to other zones.

Holes may appear if zones do not contain the minimal number of points specified with the **Minimum Points** slider in the meshing interface, for example. In this case, it is recommended to manually reactivate these zones (by right-clicking on them).

An additional cause for the appearance of holes is if meshing parameters are not appropriately selected:

- **Square Meshing** – The side length is chosen too small.
- **Voronoi Tessellation** – To many cells are chosen considering the number of localizations in the loaded file.
- **Quad-Tree Meshing** – The minimum capacity or minimum side length is chosen too small.

For large maps containing thousands of zones, an easy way to see the inactive zones is by setting the **cMax** slider to zero, deselecting the **Draw Trajectories** check button, and changing the grid color or turning it off in the **Overlay** tab of the meshing interface. Holes in the mesh are made much more evident, as shown in Figure 9.

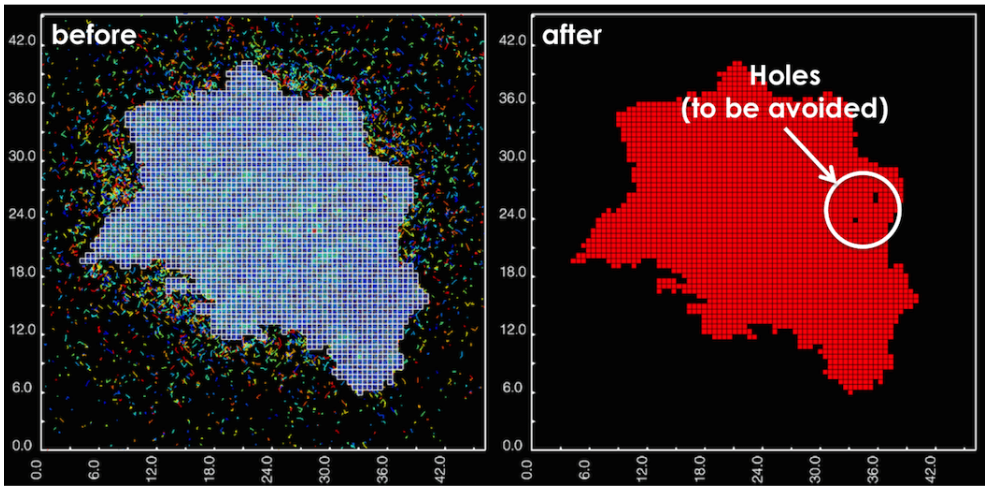


Figure 9: In adjusting visualization parameters, holes in the mesh can be made obvious. Holes such as these should be avoided (to activate the inactivate zones, it suffices to right click on them).

7 Inference

InferenceMAP uses the Bayesian inference technique that was first described in [8]. We model the motion of single particles with an overdamped Langevin equation:

$$\frac{d\vec{r}}{dt} = \frac{\vec{F}(\vec{r})}{\gamma(\vec{r})} + \sqrt{2D(\vec{r})}\xi(t) \tag{1}$$

Where $\vec{r}$ is the particle displacement vector, $\vec{F}(\vec{r})$ is the spatially-dependent force (directional bias), $\gamma(\vec{r})$ is the spatially-dependent friction coefficient or viscosity, D is the spatially-dependent diffusion coefficient, and $\xi(t)$ is a zero-average Gaussian noise term. In our case, we may model forces from a potential as $\vec{F}(\vec{r}) = -\nabla V$.

The associated Fokker-Planck equation, which governs the time-evolution of the particle transition probability $P(\vec{r_2}, t_2|\vec{r_1}, t_1)$, is given by:

$$\frac{dP(\vec{r_2}, t_2|\vec{r_1}, t_1)}{dt} = -\nabla \cdot \left(-\frac{\nabla V(\vec{r})}{\gamma(\vec{r})} P(\vec{r_2}, t_2|\vec{r_1}, t_1) - \nabla(D(\vec{r})P(\vec{r_2}, t_2|\vec{r_1}, t_1)) \right) \tag{2}$$

There is no general analytic solution to Equation 2 for an arbitrary diffusion coefficient D and potential energy V. However, if we spatially partition (mesh) the area explored by the single-particle trajectory, we may assume a constant D and V within each partition, upon which the general solution to Equation 2 is a Gaussian, described in:

$$P\left((\vec{r_2}, t_2|\vec{r_1}, t_1)|D_{i,j}, V_{i,j}\right) = \frac{\exp - \left(\frac{\left(\vec{r_2} - \vec{r_1} - \frac{\nabla V_{i,j}(t_2 - t_1)}{\gamma_{i,j}}\right)^2}{4\left(D_{i,j} + \frac{\sigma^2}{(t_2 - t_1)}\right)(t_2 - t_1)} \right)}{4\pi \left(D_{i,j} + \frac{\sigma^2}{(t_2 - t_1)}\right)(t_2 - t_1)} \tag{3}$$

Where i, j represent indices for the zones of the mesh and σ represents the experimental localization precision. An advantage to this approach is that each mesh zone is free to have a different D and V (they are not necessarily constant over the entirety of the trajectory).

The overall probability of a trajectory T due to the spatially dependent variables $D_{i,j}$ and $V_{i,j}$ is computed by multiplying the probabilities of all the individual subdomains $P(T|D_{i,j}, V_{i,j})$ to give an expression for the likelihood:

$$P\left(T|D, V\right) = \prod_{i,j} P\left(T|D_{i,j}, V_{i,j}\right) \tag{4}$$

With Equation 4 we apply Bayes' Rule, which states:

$$P\left(D, V|T\right) = \frac{P\left(T|D, V\right) P\left(D, V\right)}{P\left(T\right)} \tag{5}$$

Where $P\left(D, V|T\right)$ is the *posterior probability*, $P\left(D, V\right)$ is the *prior probability*, and $P\left(T\right)$ is the *evidence* (which is treated as a normalization constant).

For each mesh zone, we perform an optimization of the posterior probability $P\left(D, V|T\right)$ for the model parameters D, V (or $\vec{F}$). This is the *maximum a posteriori* (**MAP**) Bayesian inference approach which is used in **InferenceMAP**.

7.1 Inference Workflow

InferenceMAP offers different modes of performing the inference calculation. Selection between the different modes depends largely on the biological context and what dynamic information the user desires to have extracted. Table 1 summarizes the features of each of the inference modes.

The inference mode is selected from the **Mode** drop-down menu in the chosen meshing interface (Figure 10).

To perform the inference calculation for a mesh, the following steps are generally taken:

1. The mesh is applied by pressing the **Apply** button of the meshing interface

Table 1: Summary of different inference modes available in **InferenceMAP** .

Inference Mode	**Parameters**	**Speed**	**Priors**	**Generated Maps**
(D)	D	Fast	Uniform, Jeffreys, Smoothing	Diffusion Maps
(D,F)	D, F, V	Fast	Uniform, Jeffreys, Smoothing	Diffusion, Force and *indirect* Potential
(D,Drift)	D, $\frac{\vec{F}}{\gamma}$	Fast	Uniform, Jeffreys, Smoothing	Diffusion and Drift
(D,V)	D, F, V	Medium	Uniform, Jeffreys, Smoothing	Diffusion, Conservative Force and Potential
Polynomial Potential	D, F, V	Slow	Uniform, Jeffreys	Diffusion, Force and Confined Potential

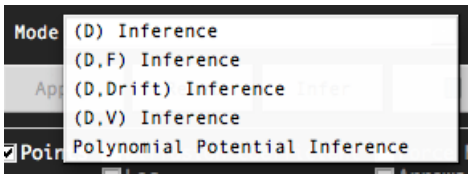


Figure 10: Inference mode selection in meshing interface.

2. Adjustments to the mesh may be made as described in Section 6.4 (e.g. activating or deactivating selected regions, choosing neighbour connections, etc.)
3. The localization precision is specified with the **Localization Precision** slider in the **Inference** tab of the meshing interface.
4. Priors may be selected or deselected in the **Prior** tab of the meshing interface.
5. Depending on the number of zones and the inference mode selected, **Randomized Optimization** may be activated in the **Advanced** tab of the meshing interface.
6. Physical parameters are inferred by pressing the **Infer** button in the **Inference** tab of the meshing interface.

7.2 (D) Inference

The **(D) Inference** mode estimates solely the diffusion coefficient in the active zones of a mesh. Diffusion is estimated in each zone independently from the others, resulting in a rapid calculation as this mode consists of a single-variable optimization. The posterior probability used to infer the diffusion in a given zone is given by:

$$P\left(\{D_{i,j}\}|\{T_k\}\right) \propto \prod_k \prod_{\mu:\vec{r}_\mu^k \in S_{i,j}} \frac{\exp\left(-\frac{\left(\vec{r}_{\mu+1}^k - \vec{r}_\mu^k\right)^2}{4\left(D_{i,j}+\frac{\sigma^2}{\Delta t}\right)\Delta t}\right)}{4\pi\left(D_{i,j}+\frac{\sigma^2}{\Delta t}\right)\Delta t} \times P_J\left(D_{i,j}\right) \times P_S\left(D_{i,j}\right) \tag{6}$$

Where D is the diffusion coefficient, μ designates the index for which the points $\vec{r}_\mu^k$ of the k^{th} trajectory are in $S_{i,j}$ (the current zone being analyzed), and σ is the experimental localization accuracy (30 nm by default).

The $P_J\left(D_{i,j}\right)$ term in (6) designates the optional Jeffreys' prior (Section 8.2). Jeffreys' prior may be activated and deactivated in the **Priors** tab of the meshing interface.

The $P_S(D_{i,j})$ term is the diffusion smoothing prior which is described in Section 8.3.

The final result of this calculation is the maximum a posteriori (MAP) estimate, $D_{i,j}^{MAP}$, and is updated to the mesh in the display window.

7.2.1 Applicability

The **(D) Inference** mode is well-suited to trajectories in the following situations:

- Freely diffusing molecules
- Rapid characterization of the diffusivity

7.3 (D,F) Inference

The **(D,F) Inference** mode estimates the diffusion coefficient and force components in the active zones of a mesh. Parameters are estimated in each zone independently from the others. Specifically, the Bayesian inference effectuated in each zone *i,j* is an optimization of D and $\vec{F}$ in the zonal posterior probability:

$$P\left(\{D_{i,j}\},\{\vec{F}_{i,j}\}|\{T_k\}\right) \propto \prod_k \prod_{\mu:\vec{r}_\mu^k \in S_{i,j}} \frac{\exp\left(-\frac{\left(\vec{r}_{\mu+1}^k - \vec{r}_\mu^k - D_{i,j}\vec{F}_{i,j}\Delta t/k_B T\right)^2}{4\left(D_{i,j}+\frac{\sigma^2}{\Delta t}\right)\Delta t}\right)}{4\pi\left(D_{i,j}+\frac{\sigma^2}{\Delta t}\right)\Delta t} \times P_J\left(D_{i,j}\right) \times P_S\left(D_{i,j}\right) \tag{7}$$

Where D is the diffusion coefficient, $\vec{F}$ is the force vector, μ designates the index for which the points $\vec{r}_\mu^k$ of the k^{th} trajectory are in $S_{i,j}$ (the current zone being analyzed), and σ is the experimental localization accuracy (30 nm by default).

The $P_J\left(D_{i,j}\right)$ term in (6) designates the optional Jeffreys' prior (Section 8.2). Jeffreys' prior may be activated and de-activated in the **Priors** tab of the meshing interface.

The $P_S(D_{i,j})$ term is the diffusion smoothing prior which is described in Section 8.3.

The final result of this calculation are the maximum a posteriori (MAP) estimates, $D_{i,j}^{MAP}$ and $\vec{F}_{i,j}^{MAP}$ and are updated to the mesh in the display window.

For further information regarding this calculation, see reference [1].

7.3.1 Potential Calculation

The user has the option of estimating the potentials, V, following the $D_{i,j}^{MAP}$ and $\vec{F}_{i,j}^{MAP}$ calculation described in Section 7.3. Potential values are estimated with a least-squares minimization between $\vec{F}_{i,j}^{MAP}$ (or $\nabla V_{i,j}^{MAP}$) and the gradient of the theoretical values for the potential under thermal equilibrium conditions, $\nabla V_{i,j}$. A user-defined penalization factor, β, is introduced to penalize the effect of strong potential gradients. A typically used value for β is ~ 2.0 (default). It is emphasized that low values of β favor large local variations in the potential field, while high values will act to damp large variations.

Specifically, the minimization is performed on zones that have at least one neighbor. The calculation minimizes *xi* as descibed in:

$$\xi\left(V_{i,j}\right) = \sum_{i,j}\left(\nabla V_{i,j} - \nabla V_{i,j}^{MAP}\right)^2 + \beta\sum\left(\nabla V_{i,j}\right)^2 \tag{8}$$

The penalization factor, β can be specified in the **Advanced** tab of the chosen meshing interface (Figure 11).

7.3.2 Applicability

The **(D,F) Inference** mode is well-suited to trajectories in the following situations:

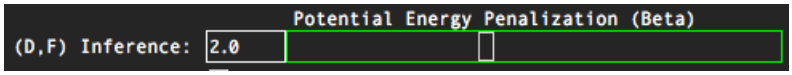


Figure 11: Potential energy penalization factor β selection slider.

- Mapping of local force components
- The presence of non-potential forces (e.g. a rotational component)
- Rapid characterization of the diffusion coefficient and directional biases of the trajectories

7.4 (D,Drift) Inference

The **(D,Drift) Inference** mode estimates the diffusion coefficient and drift ($\frac{\vec{F}}{\gamma}$) in the active zones of a mesh. Prameters are estimated in each zone independently from the others. Specifically, the Bayesian inference effectuated in each zone *i,j* is an optimization of D and $\frac{\vec{F}}{\gamma}$ in the zonal posterior probability:

$$P\left(\{D_{i,j}\},\{\frac{\vec{F}_{i,j}}{\gamma}\}|\{T_k\}\right) \propto \prod_k \prod_{\mu:\vec{r}_\mu^k \in S_{i,j}} \frac{\exp\left(-\frac{\left(\vec{r}_{\mu+1}^k - \vec{r}_\mu^k - \frac{\vec{F}_{i,j}}{\gamma}\Delta t/k_B T\right)^2}{4\left(D_{i,j}+\frac{\sigma^2}{\Delta t}\right)\Delta t}\right)}{4\pi\left(D_{i,j}+\frac{\sigma^2}{\Delta t}\right)\Delta t} \times P_J\left(D_{i,j}\right) \times P_S\left(D_{i,j}\right) \tag{9}$$

Where D is the diffusion coefficient, $\vec{F}$ is the force vector, γ is the friction (viscosity), μ designates the index for which the points $\vec{r}_\mu^k$ of the k^{th} trajectory are in $S_{i,j}$ (the current zone being analyzed), and σ is the experimental localization accuracy (30 nm by default).

The $P_J\left(D_{i,j}\right)$ term in (6) designates the optional Jeffreys' prior (Section 8.2). Jeffreys' prior may be activated and deactivated in the **Priors** tab of the meshing interface.

The $P_S(D_{i,j})$ term is the diffusion smoothing prior which is described in Section 8.3.

The final result of this calculation are the maximum a posteriori (MAP) estimates, $D_{i,j}^{MAP}$ and $\frac{\vec{F}}{\gamma}_{i,j}^{MAP}$ and are updated to the mesh in the display window.

7.4.1 Applicability

The **(D,Drift) Inference** mode is well-suited to trajectories in the following situations:

- Active processes (e.g. active transport phenomena)

7.5 (D,V) Inference

The **(D,V) Inference** mode directly computes the diffusion coefficient and the gradients of the potential in each active zone in the mesh. Naturally the calculation is expensive as it estimates all the variables (the diffusion coefficients and potential energies) in all the zones of the mesh at the same time. A randomized optimization method (Section 7.7) is included to alleviate its generally high computational requirements. The zonal posterior probability is described as:

$$P\left(\{D_{i,j}\},\{\nabla V_{i,j}\}|\{T_k\}\right) \propto \prod_k \prod_{\mu:\vec{r}_\mu^k \in S_{i,j}} \frac{\exp\left(-\frac{\left(\vec{r}_{\mu+1}^k - \vec{r}_\mu^k - D_{i,j}\nabla V_{i,j}\Delta t/k_B T\right)^2}{4\left(D_{i,j}+\frac{\sigma^2}{\Delta t}\right)\Delta t}\right)}{4\pi\left(D_{i,j}+\frac{\sigma^2}{\Delta t}\right)\Delta t} \times P_J\left(D_{i,j}\right) \times P_S\left(D_{i,j},V_{i,j}\right) \tag{10}$$

Where D is the diffusion coefficient, V is the potential energy, μ designates the index for which the points $\vec{r}_\mu^k$ of the k^{th} trajectory are in $S_{i,j}$ (the current zone being analyzed), and σ is the experimental localization accuracy (30 nm by default).

The $P_J\left(D_{i,j}\right)$ term in (6) designates the optional Jeffreys' prior (Section 8.2). Jeffreys' prior may be activated and deactivated in the **Priors** tab of the meshing interface.

The $P_S(D_{i,j}, V_{i,j})$ term is the diffusion and potential smoothing prior which is described in Section 8.3.

7.5.1 Applicability

The **(D,V) Inference** mode is well-suited to trajectories in the following situations:

- Systems with stable interaction sites

7.6 Polynomial Potential Inference

In contexts where there is clear confinement of a tracked particle, it is useful to describe the confining energy as a polynomial of order C in:

$$V(\vec{r}) = \sum_{j=0}^{C} \sum_{i=0}^{j} \alpha_{i,j} x^i y^{j-i} \tag{11}$$

Where the constants $\alpha_{i,j}$ are fitted to the experimental force fields using standard simplex methods.

This description of the potential is integrated into the inference calculation as in Equation 7. This type of inference is prohibitively expensive for large maps, it is recommended for localized trajectories in small regions. Further information regarding this type of modelling is described in reference [8].

7.6.1 Applicability

The **Polynomial Potential Inference** mode is well-suited to trajectories in the following situations:

- Confined trajectories

7.7 Randomized Optimization

To tackle large problems consisting of several hundred or thousand of individual zones, a **Randomized Optimization** function is available, which greatly reduces computation time. It can be used in the following inference modes:

- **(D) Inference** when a smoothing prior is used
- **(D,F) Inference** when a smoothing prior is used
- **(D,Drift) Inference** when a smoothing prior is used
- **(D,V) Inference**

This feature is available for all meshing modes, parameters of which can be adjusted in the **Advanced** tab of the meshing interface, seen in Figure 7.7.

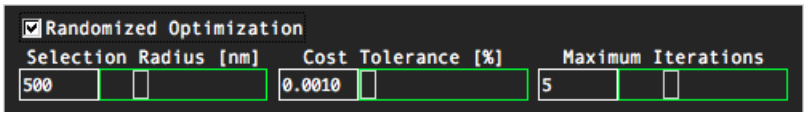


Figure 12: **Randomized Optimization** options in the meshing interface.

To activate the **Randomized Optimization** mode, a compatible mode must be selected in the **Inference** tab of the meshing interface (the smoothing prior also needs to be activated for the cases of **(D)**, **(D,F)**, and **(D,Drift)** inference modes). Activation consists in clicking the **Randomized Optimization** check box seen in Figure 7.7.

This function works as follows:

- Subregions of zones are selected in a circle of radius defined by the **Selection Radius [nm]** slider (Figure 7.7). The zone in which the circle is centered is selected randomly among the activated zones of the mesh. It is advisable to select a radius such that roughly ∼10 zones or more will be encompassed.

- The inference calculation is performed *only* for the subregion zones, parameters (e.g. diffusion) in all other zones remaining constant. Calculations are limited to the number of iterations specified in the **Maximum Iterations** slider (Figure 7.7). It is recommended to keep the maximum number of iterations between 5–10. Parameter values for the subregion zones are updated after the maximum number of iterations has been reached, thereafter another selection circle is chosen.
- A stopping condition may be specified based on the percent error between consecutive posteriori cost function values (defined in the **Cost Tolerance [%]** slider). It is recommended to keep this tolerance below 1 % as the stopping condition may be met before all zones in the mesh have been visited. Often, it is suggested to define a 0 % tolerance, and to manually stop the calculation by pressing the **Pause** button in the **Inference** tab when the cost function has decayed sufficiently, as in Figure 13. For especially large meshes, stopping the calculation will be delayed – in these cases the user is advised to press the **Pause** button, keep the cursor hovering above, and wait!

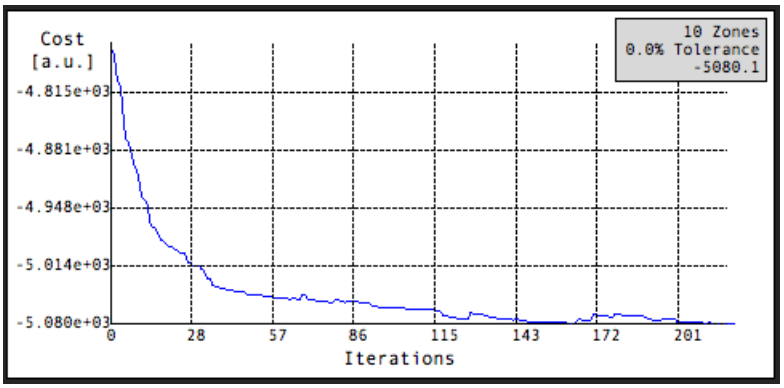


Figure 13: **Randomized Optimization** interface, demonstrating the decay in the cost function. Generally, the **Randomized Optimization** calculation may be stopped when the cost function has sufficiently decayed, as indicated.

7.8 Freehand Selection Inference

For determination of the diffusion coefficient and force components in a localized region of a trajectory overlay, **InferenceMAP** includes a **Freehand Selection** tool in the main interface, shown in Figure 14. This is useful in cases where only subregions of the trajectory space need to be analyzed.

To select a freehand region, the **Make Selection** button is pressed, whereafter the user can select (in lasso style) a closed region of the trajectory overlay by pressing and holding the left mouse button. Once a region has been circled, the user releases the left mouse button (which will close the selected region). In the **Freehand Panel** (Figure 14) the user may specify the **Localization Precision**. Afterwards, the diffusion coefficient and force or drift components determined and displayed by pressing the **Infer (D,F)** or **Infer (D,Drift)** buttons, respectively. The inferred region is highlighted in white, with an overlaid arrow to represent the angle of the force or drift (the size of the arrow does not correspond to the magnitude of the force or drift).

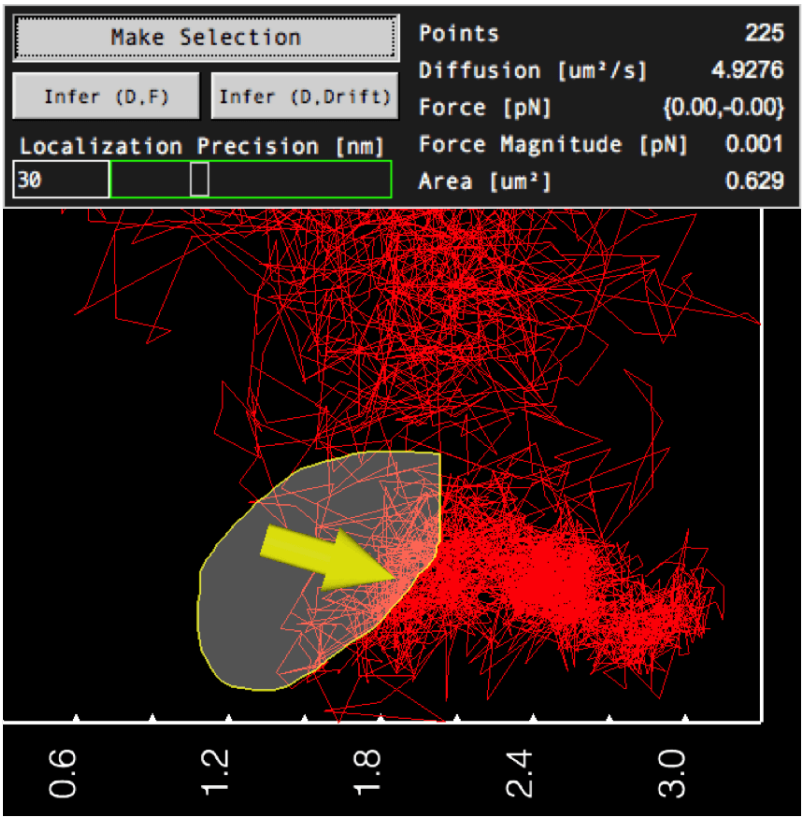


Figure 14: **Freehand Panel** (top) with selected and inferred region (bottom).

8 Priors

In general, prior probabilities are used to impose our beliefs to the dynamic parameter estimates that **InferenceMAP** performs when generating maps. This section describes three types of prior probabilities that users may incorporate in the inference calculation. They are enabled via the **Priors** tab of the meshing interfaces, shown in Figure 15.

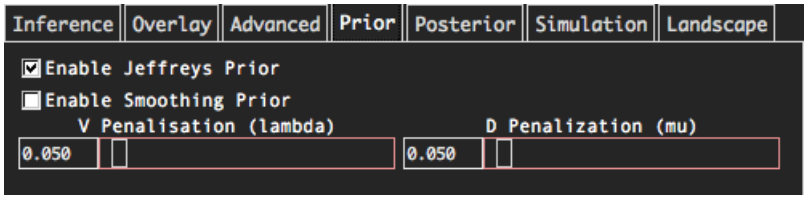


Figure 15: **Prior** tab of the meshing interface.

8.1 Uniform

If no information is known regarding the parameters being inferred (e.g. the diffusivity), the Uniform Prior should be used. It is applied by default if no other priors are activated. Specifically, use of this prior means that the results yielded from the Bayesian inference calculation are identical to those that would be obtained from a maximum likelihood estimation.

8.2 Jeffreys

Jeffreys prior is used to ensure that the posterior probability distribution of an inference calculation is invariant by re-parameterization. Moreover, it allows "protecting" inference of the diffusion in cases of high local confinement. In such situations, the "effective diffusion" introduced by a non-zero positioning noise can lead to inference of a negative diffusion value [1]. We emphasize that without positioning noise the inference never leads to negative diffusion. Table 2 lists forms of Jeffreys prior for the different inference modes.

Jeffreys prior is applicable to a large number of situations. Situations where it is not recommended are in cases where the diffusion is similar to the effective "noise diffusion" which we estimate as $\frac{\sigma^2}{\Delta t}$. It may be activated and deactivated in the **Prior** tab of the meshing interface.

Table 2: Forms of Jeffreys' prior for the different inference modes.

Inference Mode	**Jeffreys Prior,** P_J
(D)	$\frac{1}{(D_{i,j}\Delta t+\sigma^2)}$
(D,F)	$\frac{D_{i,j}^2}{(D_{i,j}\Delta t+\sigma^2)^2}$
(D,Drift)	$\frac{1}{(D_{i,j}\Delta t+\sigma^2)^2}$
(D,V)	$\frac{D_{i,j}^2}{(D_{i,j}\Delta t+\sigma^2)^2}$
Polynomial Potential	$\frac{D_{i,j}^2}{(D_{i,j}\Delta t+\sigma^2)^2}$

An important remark is that in the cases of **D,F** and **D,V** inference, Jeffreys' prior prevents inferrring negative diffusion coefficients, even when position noise is high.

8.3 Smoothing

The smoothing prior penalizes gradients of the physically inferred parameters (either the diffusion or the potential energy, depending on the inference mode). This prior is appropriate for use in biological systems where notions of the strength of

diffusion coefficient and potential energy gradients exist. It is meant to reinforce the physical behavior that is to be expected in certain biological systems. For example, in certain situations we do not expect large jumps zone-to-zone in the diffusion coefficient.

The diffusion smoothing prior is defined by the surface integral in Equation 12. The coefficient μ is the diffusion gradient penalization factor which modulates the strength of smoothing in the area S.

$$P_S\left(D_{i,j}\right) = \exp\left(-\mu \iint\limits_s ||\nabla D_{i,j}||^2 ds\right) \tag{12}$$

The potential smoothing prior is defined by the surface integral in Equation 13. The coefficient λ is the potential gradient penalization factor which modulates the strength of smoothing in the area S.

$$P_S\left(V_{i,j}\right) = \exp\left(-\lambda \iint\limits_s ||\nabla V_{i,j}||^2 ds\right) \tag{13}$$

Values for μ and λ are specified in the **Prior** tab of the meshing interface as in Figure 15. Table 3 lists the types of smoothing priors available for the different inference modes.

Table 3: Forms of smoothing priors for the different inference modes.

Inference Mode	**Smoothing Prior,** P_S
(D)	Diffusion
(D,F)	Diffusion
(D,Drift)	Diffusion
(D,V)	Diffusion, Potential
Polynomial Potential	N/A

Although the **(D)**, **(D,F)**, are **(D,Drift)** modes are generally rapid calculations, the computation time increases substantially when the diffusion smoothing prior is activated. The reason for this is that the problem becomes a calculation in which parameters in all zones (e. g. diffusion, force, drift) are being optimized simultaneously (without the smoothing prior parameters are optimized in each zone independently which greatly reduces the dimensionality of the problem). For this reason, it may be of interest to use a **Randomized Optimization** (Section 7.7) in large mapping problems which utilize smoothing priors.

9 Posterior Sampling

As explained in Section 7, the maximum value of the *posterior probability* distribution (the MAP) is used to estimate the dynamic parameters (diffusion coefficient, force components, drift components, and potential energy). Sampling of the *posterior probability* (see Section 7) gives insight into the precision of the estimations. Typically, the *posterior probability* distribution takes the approximate form of a Gaussian, where the full-width at half-maximum may be calculated to measure variance of the estimation.

InferenceMAP offers the possibility to sample posterior values for the different parameters. For each of the different meshing types, *after* the inference calculation, the user has the option to sample to posterior probability for the different parameters. Posteriors are sampled for different zones, enabling the user to compare the parameter estimation precision in different parts of the mesh. The different inference modes impose constraints on which parameters may be sampled, as Table 4 shows.

Table 4: Availability of posterior probability sampling for different parameters for different inference modes.

Inference Mode	Diffusion Coefficient	Force Magnitude	Drift Magnitude	Potential Energy
(D)	Yes	No	No	No
(D,F)	Yes	Yes	No	No
(D,Drift)	Yes	No	Yes	No
(D,V)	Yes	No	No	Yes
Polynomial Potential	N/A	N/A	N/A	N/A

After the inference calculation, posterior probabilities are sampled in the **Posterior** tab of the meshing interface, shown in Figure 16. The steps to sampling the posterior are the following:

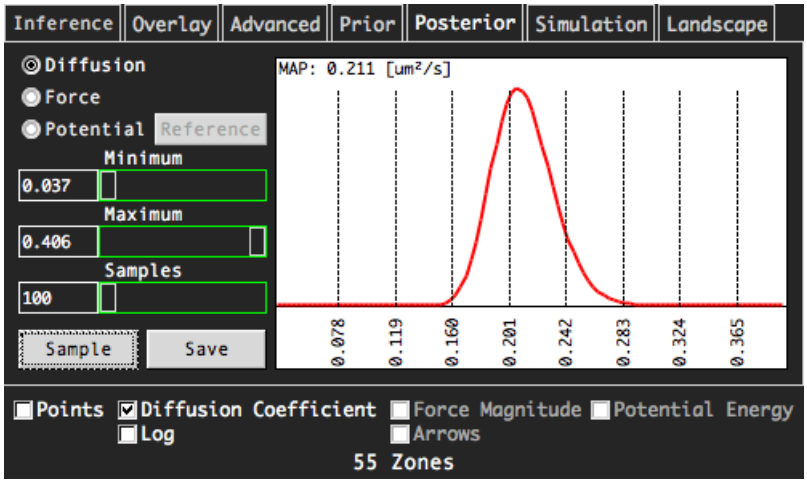


Figure 16: **Posterior** tab in the meshing interface.

1. A zone in the mesh in which to sample the posterior is selected with the mouse in the **Display Window**
2. The estimation parameter (diffusion coefficient, force magnitude, or potential energy) is selected in the top-left of the tab
3. Sampling ranges are selected via the **Minimum** and **Maximum** sliders
4. The number of (equally-spaced) samples are selected with the **Samples** slider
5. The posterior for the chosen parameter in the selected zone is sampled by pressing the **Sample** button, a trace of which is displayed in the interface
6. Posterior sampling data may be exported to a simple ASCII file by pressing the **Save** button

For the **(D,V) Inference** mode, the posterior of the potential may be sampled in two ways:

1. A single-zone of the mesh can have its potential sampled across a user-defined range
2. The posterior of the difference of potentials between two zones may be sampled

For sampling the posterior of the difference of potentials, the first zone is selected with the mouse in the **Display Window**. Following, by pressing the **Reference** button, the second may be selected (see Figure 17). At this point, the user may select the range of potential differences to sample between the two zones (specified with the **Maximum** slider). This type of calculation is useful for measuring the precision of potential energy barriers, for example.

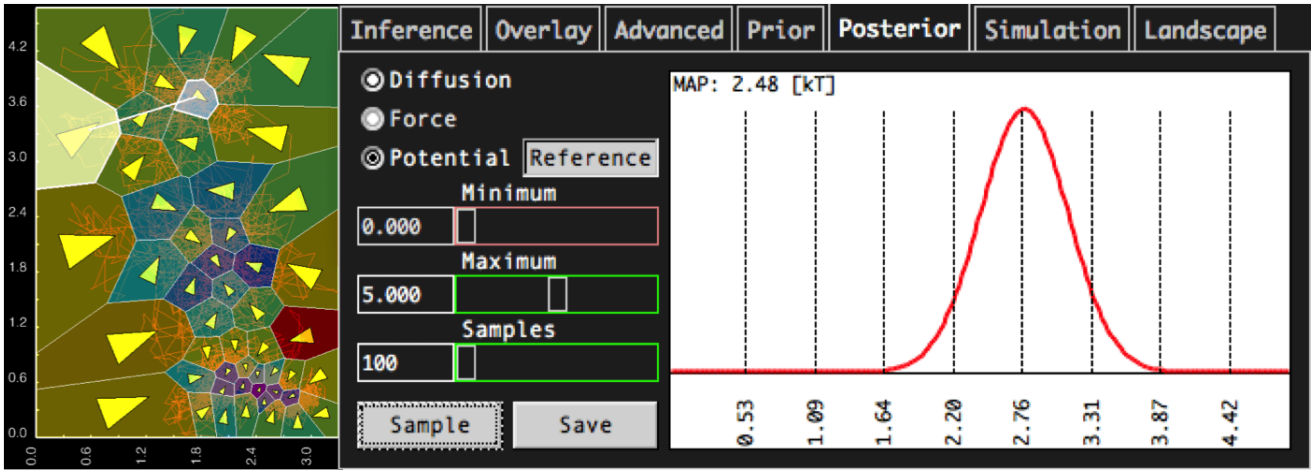


Figure 17: Posterior sampling of difference of potentials in two zones.

10 Trajectory Simulation

After an inference calculation has been performed to generate diffusion and potential energy maps, there is the possibility to generate simulated trajectories based on the inferred parameter values. This is based on the Gillespie method, described in [1].

The user may be motivated to create simulated trajectories in a few different situations. Namely, in cases where trajectories are short (e.g. less than 10 points) and certain "long trajectory" metrics would like to be estimated. These may include measuring whether a trajectory experiences anomalous motion, residence time, and binding and dissociation rates.
Simulated trajectories are generated via the **Simulation** tab of the meshing interface, seen in Figure 18.

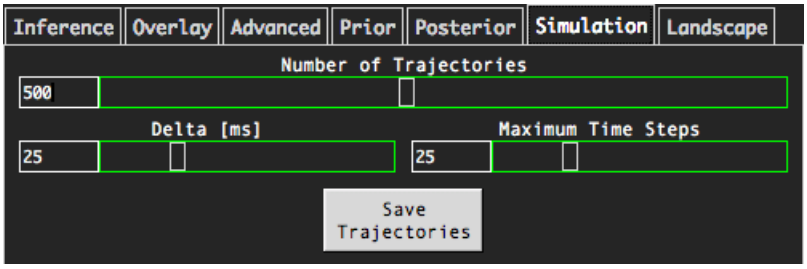


Figure 18: **Simulation** tab of the meshing interface.

In this tab, the user may select the number of trajectories, the time spacing (delta) between consecutive points, and the maximum time steps in an individual trajectories. Pressing the **Save Trajectories** button outputs all trajectories into a **.trxyt** file (described in Section 3). Output trajectories will have a spatial resolution corresponding to that of the mesh.

11 Landscape Viewing

InferenceMAP offers to option to view dynamical maps as three-dimensional landscapes for more intuitive interpretation. This feature is accessed with the **Landscape** tab of the meshing interface, shown in Figure 19.

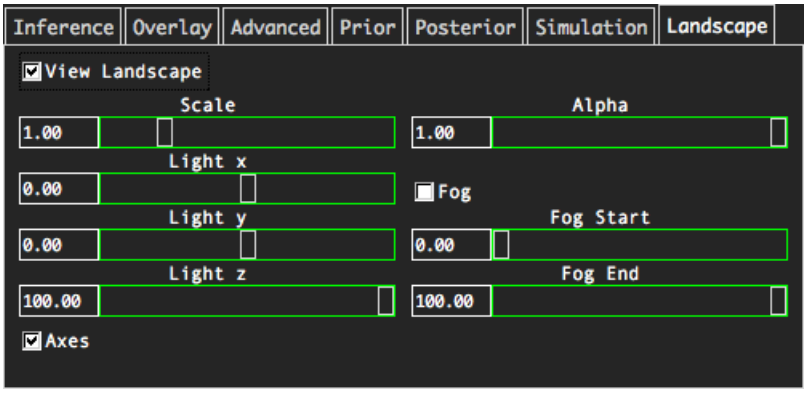


Figure 19: **Landscape** tab of the meshing interface.

Here, the user has various visualization options for viewing the three-dimensional landscape. To change between landscapes, simply press the parameter boxes on the button of the meshing interface. The landscape viewing mode is available for all meshing types (see Section 6). Color scales can be adjusted with the **cMin** and **cMax** sliders in the **Main Interface**. Figure 20 shows the three-dimensional landscapes for the same trajectory file (a simulated potential well).

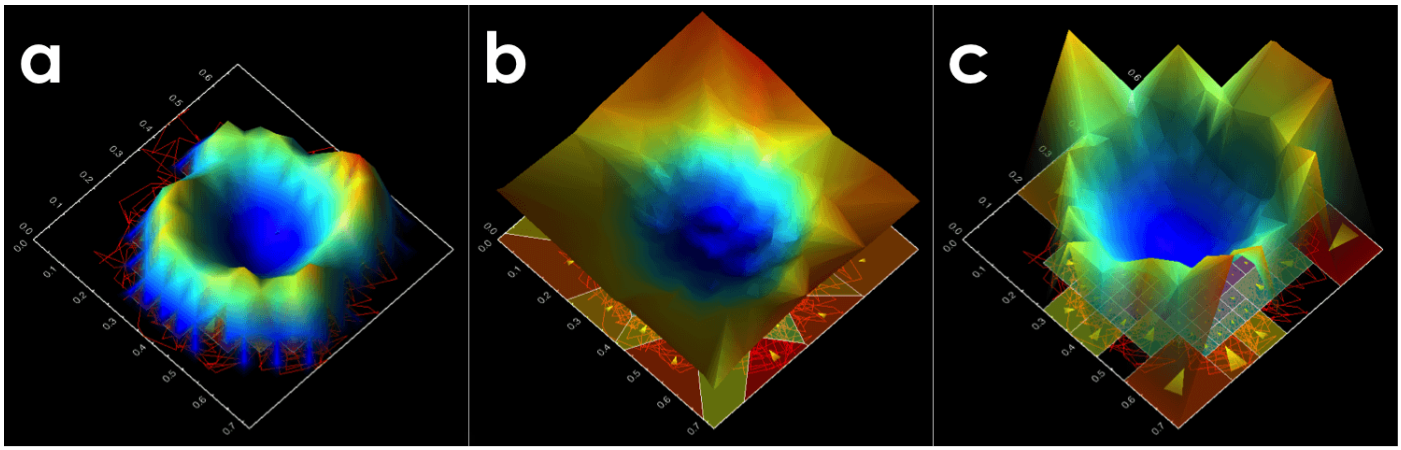


Figure 20: Potential energy landscapes of the same trajectory file (a simulated potential well) for the different meshing types: (a) square mesh, (b) Voronoi tessellated mesh, and (c) a quad-tree mesh.

12 Freehand Selection

Often it is desired to only generate a dynamical map for a subregion of the entire trajectory space. **InferenceMAP** enables users to select custom subregions with the freehand selection tool, by selecting the **Make Selection** button in the **Freehand Panel** of the main interface window (top panel of Figure 14), whereafter the user selects the desired subregion in the **Display Window** by clicking and holding the right mouse button. Once a closed region has been selected, releasing the right mouse button will create the subregion. At this point, the user opens a meshing interface as usual, by choosing from the options in the **Inference** $>$ **Meshing** menu of the **Main Menu**. The mesh generated in the subregion upon pressing the **Apply** button in the respective meshing interface. Figure 21 shows an example of a map for a custom-selected region using this feature.

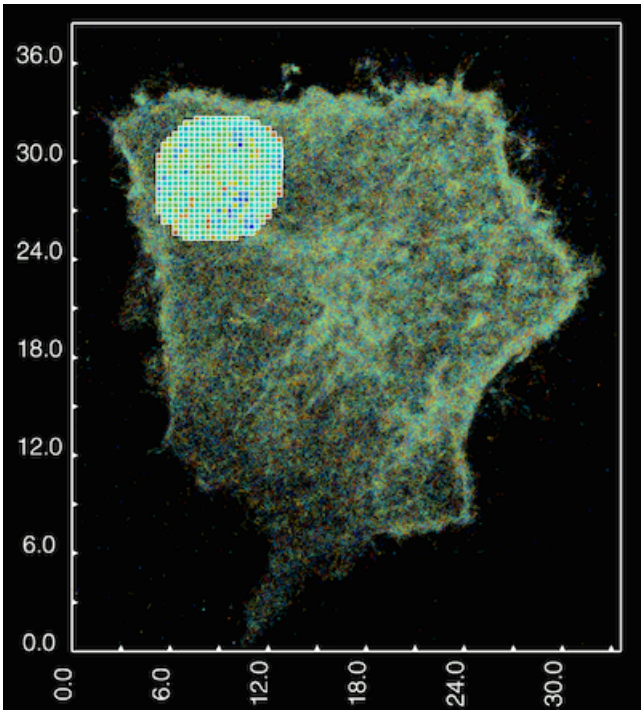


Figure 21: Example of a map calculated for a custom-selected subregion.

It is important to assure that the **Make Selection** button is toggled *on* before pressing **Apply** in the chosen meshing interface.

13 Tools

13.1 Annotations

Various annotations may be adjusted or removed in the **Display Window** by accessing the **Annotations** interface in the **Tools** menu of the **Main Menu**, seen in Figure 22.

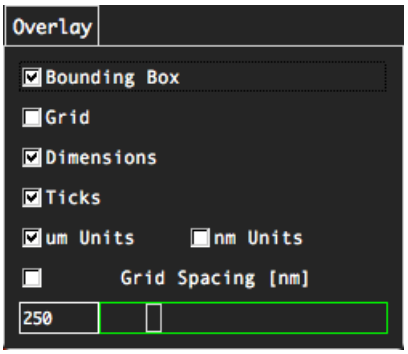


Figure 22: **Annotations** interface.

13.2 Density Calculation

The *relative* density of localizations can be calculated by pressing the **Calculate Density** button in the **Density Panel** of the **Main Interface**. The density is determined by counting the number of localizations (neighbors) within the circle defined at each given localization. The size of this circle is determined by the radius indicated in the **Neighborhood Radius Slider Density Panel** (Figure 23).

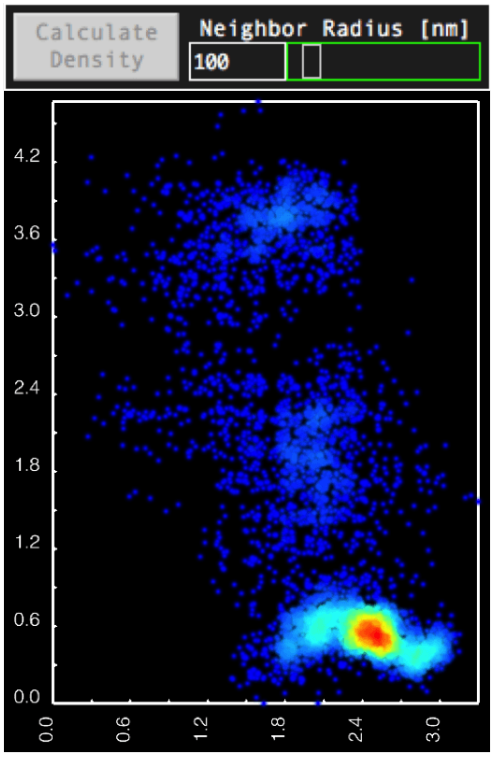


Figure 23: **Density Panel** (top) in the **Main Interface**, with localizations density plot (below).

13.3 Interval Selection

Specific time intervals in a loaded trajectory dataset can be selected for analysis. This is useful if a temporally windowed analysis is desired. The **Start Time Slider** and **End Time Slider** in the **Main Interface** are used to select the desired interaval. Only trajectories in the selected window will be updated to the display, as shown in Figure 24. Afterwards, a mesh may be applied as usual (Section 6), but only trajectories within the selected interval will be considered. Of note, the interval selection is disabled once a mesh has been applied.

Figure 24: Interval selection sliders in the **Main Interface**.

13.4 Save Screen

At any moment while using **InferenceMAP** , a screen shot of the **Display Window** may be taken by selecting **Save Screen** in the **Tools** menu of the **Main Menu**. This feature will also save a screen shot of the colorbar in a separate file (with **_colorbar** postfix).

13.5 TIFF Overlay

An important feature in **InferenceMAP** is the ability to overlay experimentally-acquired TIFF images to corresponding trajectories. Both single-image files (e.g. DIC or transmission images) and multi-image stacks (such as the raw images from which the trajectories are constructed) may be overlaid to open trajectory files (described in Section 3). A TIFF image can be overlaid to an open trajectory file, by selecting the **TIFF Overlay** button in the **Tools** menu in the **Main Menu**, shown in Figure 25. Of note, TIFF images must be 16–bit black and white to be properly overlaid.

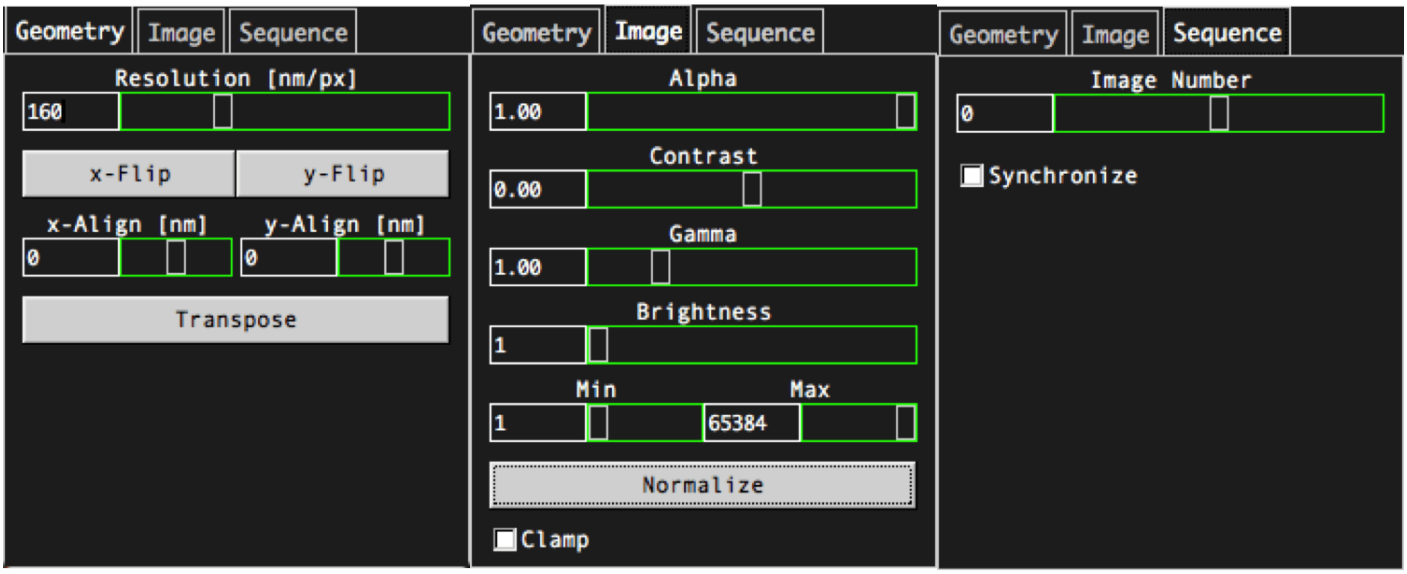


Figure 25: **TIFF Overlay** interface tabs.

The **Geometry** tab permits adjustment of the pixel size, and allows adjustments to image orientation.

The **Image** tab contains functions for image rendering. The **Clamp** box clamps the images to the maximal dimensions of the loaded trajectory file.

The **Sequence** tab applies to overlaid multi-image stacks. The **Image Number** slider allows traversal of the images in the stack. The **Sync** button synchronizes images loaded in the stack to trajectory animations (activated by clicking the **Animate Trajectories** button in the **Visualization Panel** of the **Main Interface**.

13.6 White Background

The user may change the default black background of the **Display Window** to white by toggling the **View** > **White Background** option.

14 Performance

This section outlines the performance and robustness of the inference technique subject to changes in calculation parameters. We emphasize that default calculation parameters are largely sufficient to yield accurate estimates of dynamic parameters from trajectories. For more extensive simulations and discussion of the performance of the technique, it is recommended that the user consult previous papers discussed in the **References** section.

To give an overview of the performance, we use the diffusion as our test parameter. Other inferred parameters such as the forces, potential energy, and drift follow similar trends.

14.1 Trajectory Length

As discussed in Section 7, our trajectory inference technique does not depend on the length of trajectories. Equations 6, 7, 9, and 10 clearly show that the posterior probability distribution is dependent on trajectory translocations (i.e., $\vec{r}^{k}_{\mu+1} - \vec{r}^{k}_{\mu}$) and not trajectory lengths. Effectively, the "long trajectory" likelihood is simply the product of the individual likelihoods of each translocation.

14.2 Number of Localizations

We show the effect of the number of localizations (translocations, effectively) on the inferred diffusion coefficient by sampling the posterior probability of the diffusion for increasingly long trajectories. This is entirely analogous to analyzing the posterior probability in a single zone of a mesh. This treatment is valid as per the logic of the previous section, as our technique uses individual translocations for parameter inference (i.e. it "sees" a long trajectory as a sum of two-point trajectories).

Figure 26 shows the form of the posterior for five different trajectories with different numbers of points (steps). Qualitatively, the distributions narrow with the increased number of points.

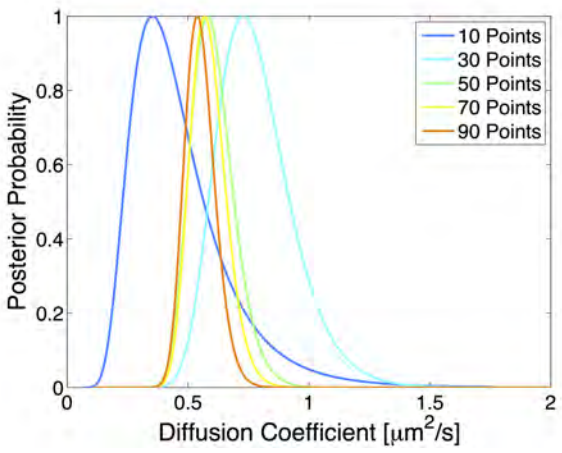


Figure 26: Plots of posterior probabilities of the diffusion coefficient as a function of number of points (translocations) in the parameter estimation.

The Bayesian inference estimator that we use is unbiased and we illustrate this in Figure 27. Here, the nominal diffusion coefficient (0.5 $\mu m^2/s$) for trajectories of different numbers of translocations is estimated. Correspondingly, the error in the estimation is embodied by its standard deviation (stdev) which is described with a $1/N$ profile. This is an important result, as the Bayesian inference method employed by **InferenceMAP** is unbiased regardless of the number of translocations per trajectory, or analogously, translocations per zone of a generated map.

14.3 Mode Selection

InferenceMAP offers numerous modes to perform inference calculations to generate maps of dynamic parameters. This sections seeks to compare the **(D) Inference** and **(D,V) Inference** modes for two simulated datasets: one in which trajectory motion is purely diffusive, and another in which an interaction region is present. Effectively, two models of motion are being compared, and depending on which is selected key physical properties may be concealed. Results of the aforementioned

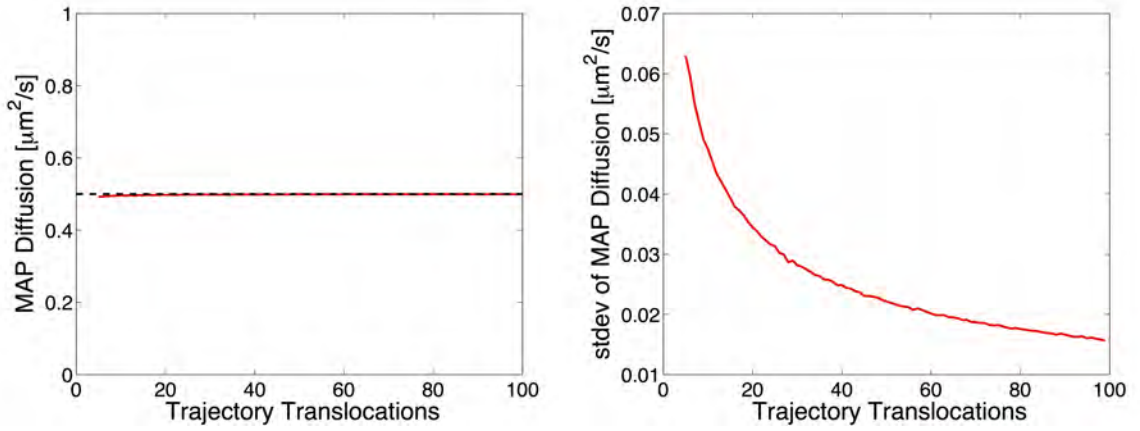


Figure 27: (left) Maximum a posteriori (MAP) estimator of diffusion coefficient on trajectories of increasing number of translocations (nominal diffusion of 0.5 $\mu m^2/s$). (right) Standard deviation of the MAP estimator.

comparison are displayed in Figure 28.

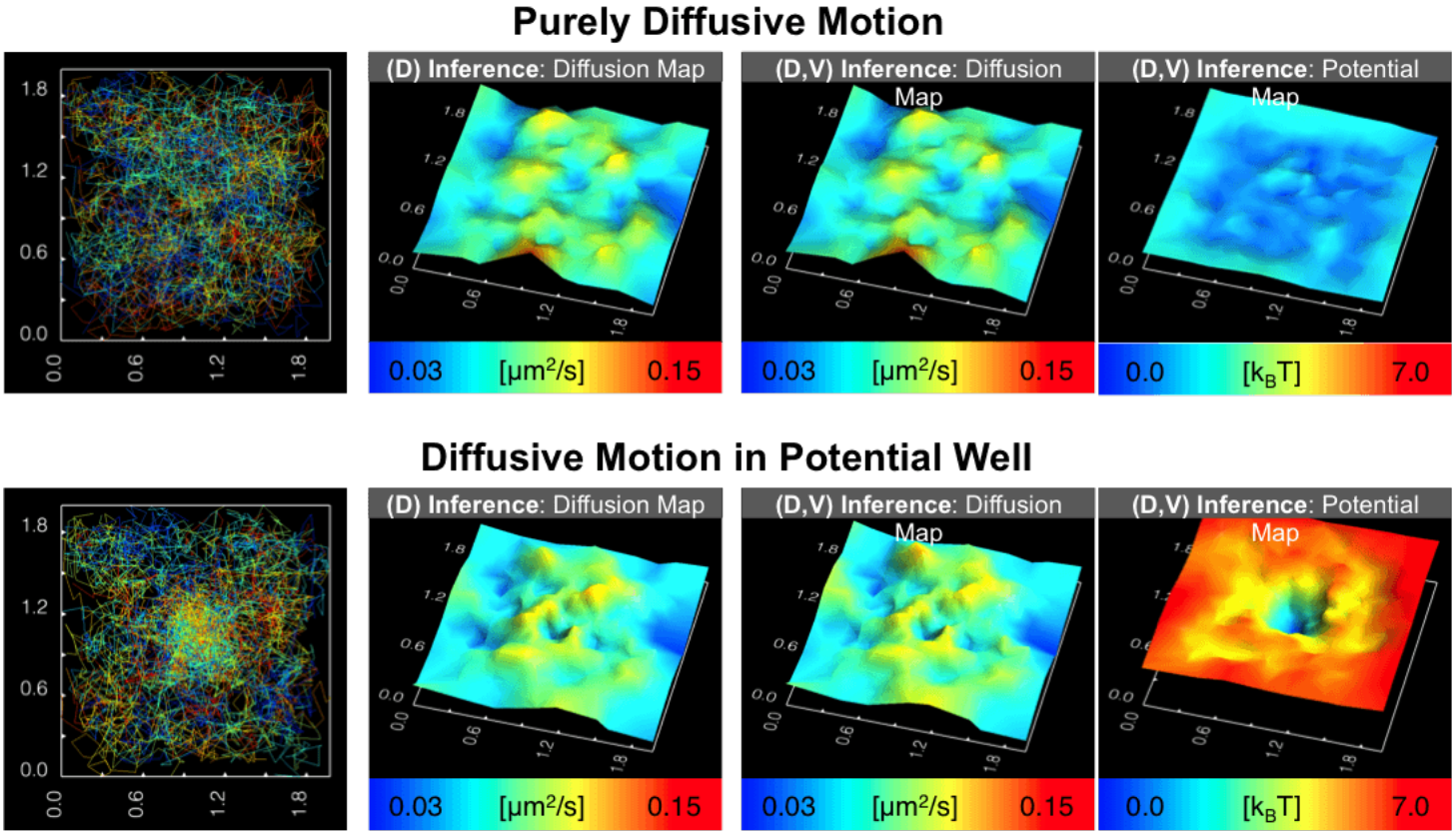


Figure 28: Comparison of different parameter maps inferred for the **(D) Inference** and **(D,V) Inference** modes.

The above examples clearly show that in the absence of an interaction energy (top row of Figure 28), the corresponding potential energy map inferred using the **(D,V) Inference** mode is essentially flat, as is expected to be the case. Conversely, the **(D) Inference** mode does not model interaction energy, and is hence not capable of revealing a large interaction region (bottom row of Figure 28).

14.4 Prior Probabilities

InferenceMAP offers two types of prior probabilities. Jeffreys' prior is recommended in many cases. It is used to ensure that the posterior probability distribution of an inference is invariant by reparameterization. Moreover it protects the inference of diffusion in cases of high local confinement.

The smoothing prior is used to penalize gradients in the inferred parameters between neighboring zones. To demonstrate the effect of the smoothing prior, we apply it to the case of a simulated trajectory set with different diffusion step gradients. Accordingly, the effect of the diffusion smoothing prior (i.e. the μ parameter described in Section 8.3) on the diffusive map is illustrated in Figure 29.

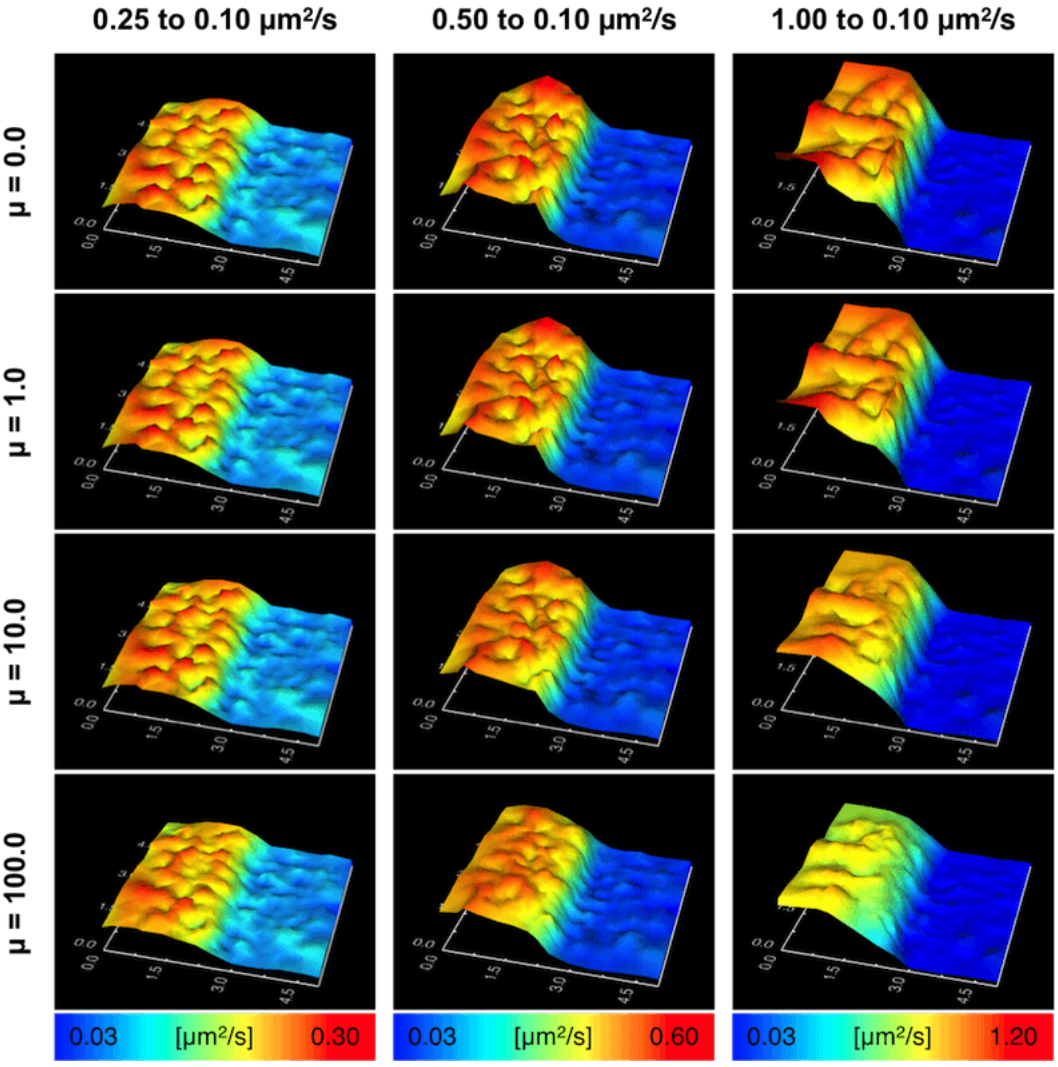


Figure 29: Table illustrating the effect of the diffusion map smoothing prior (based on the coefficient μ) on simulated trajectories. The trajectory space has two diffusive populations separated by a "step" gradient of different sizes in the different columns.

Increasing the value of μ smooths the interface between the two diffusive regions to different degrees. This prior is useful in cases in which large gradients in the diffusion are not to be expected. This is analogous to the use of the potential energy prior (i.e. the λ parameter described in Section 8.3).

15 Stepped-Through Examples

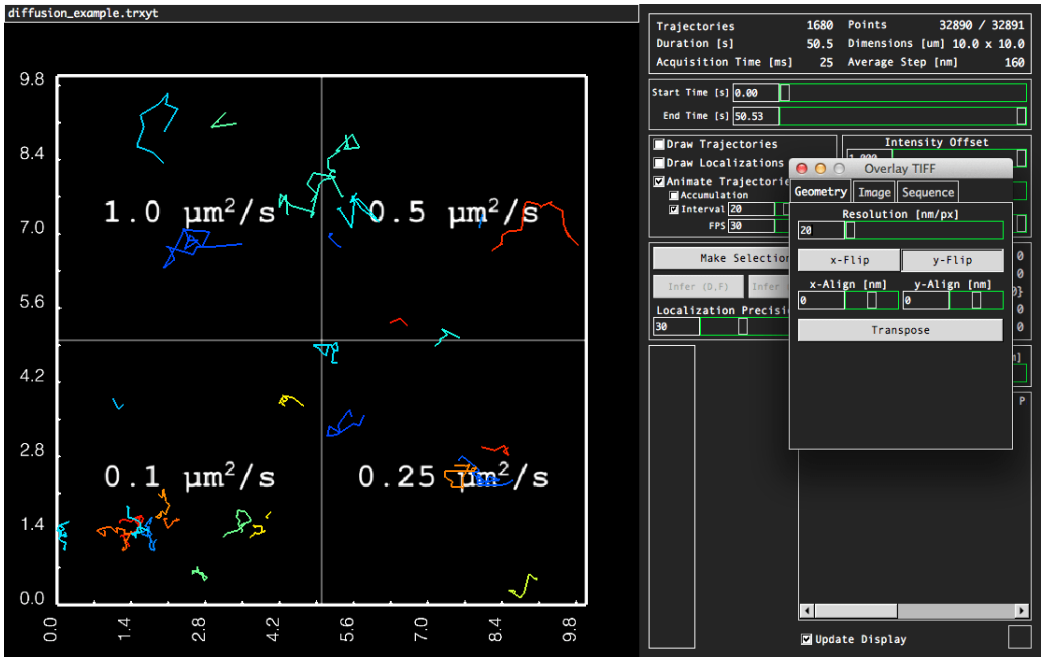
In this section, a few examples for the different inference modes in **InferenceMAP** are stepped through in detail. All the examples are used with the included *example* files, accessible from the **File** menu of the **Main Menu**.

15.1 Example 1: Generating a Diffusion Map

This example demonstrates how to generate a diffusion map of simulated molecule trajectories with realistic diffusivities. The trajectory file is partitioned into four quadrants, each corresponding to a different specified diffusion coefficient. The localization precision (positioning noise) is specified to be 30 nm.

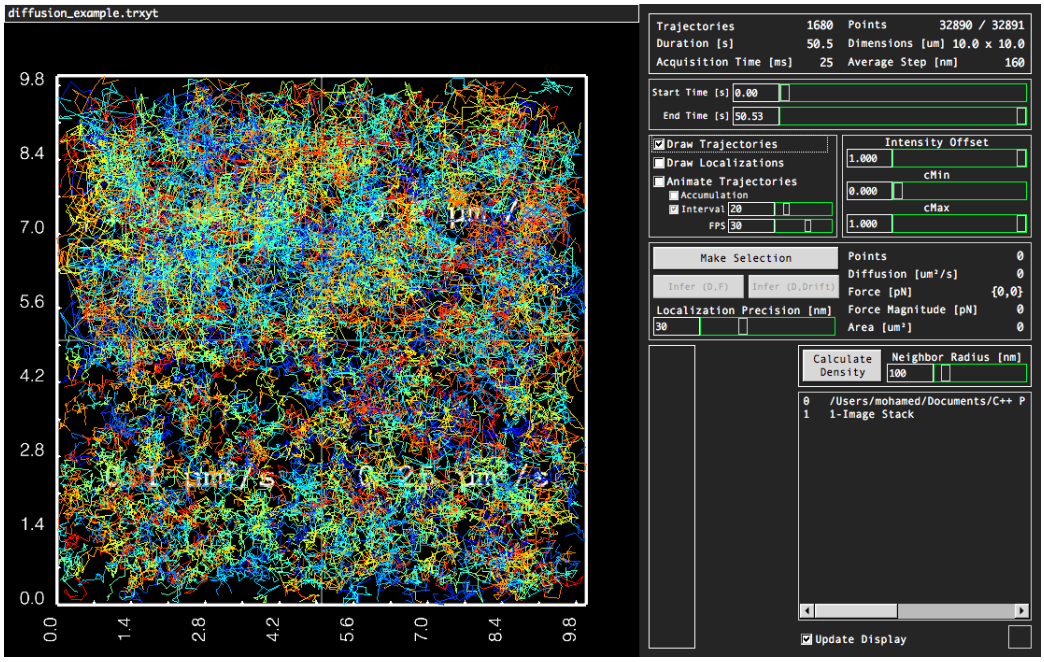
1. Load Diffusion Example File

Select the **Diffusion Example (Simulation)** file from the **File** > **Examples** menu. The trajectories will start animating. The specified diffusion coefficient for the respective quadrants is overlaid as a background image.



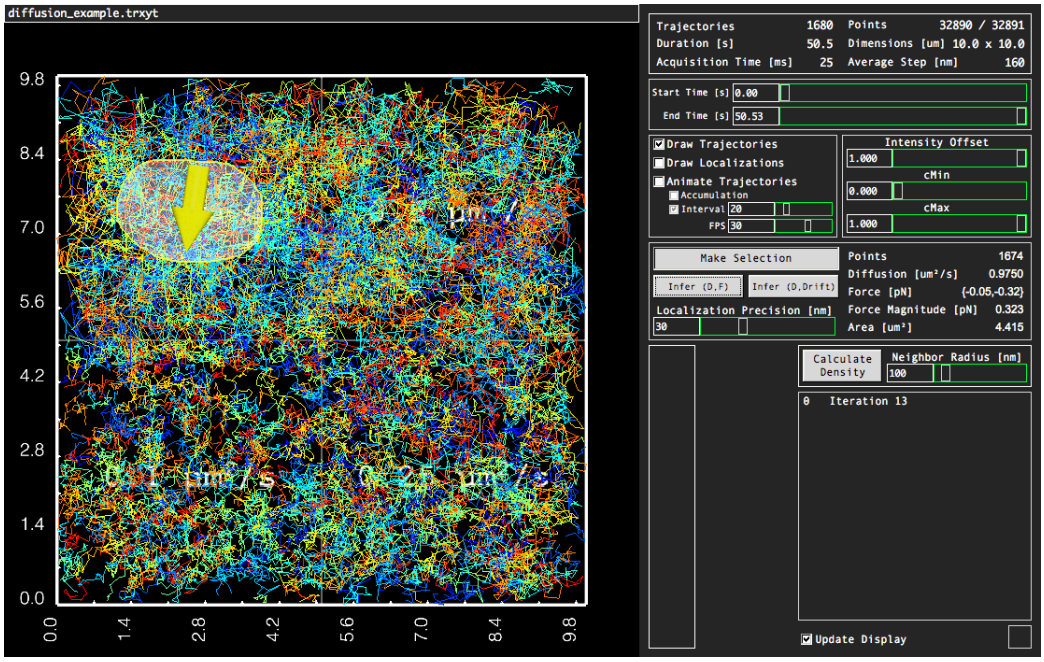
2. Visualize Trajectories

Deselect the **Animate Trajectories** button, and select the **Draw Trajectories** button in the **Main Interface** to overlay all the trajectories in the file.



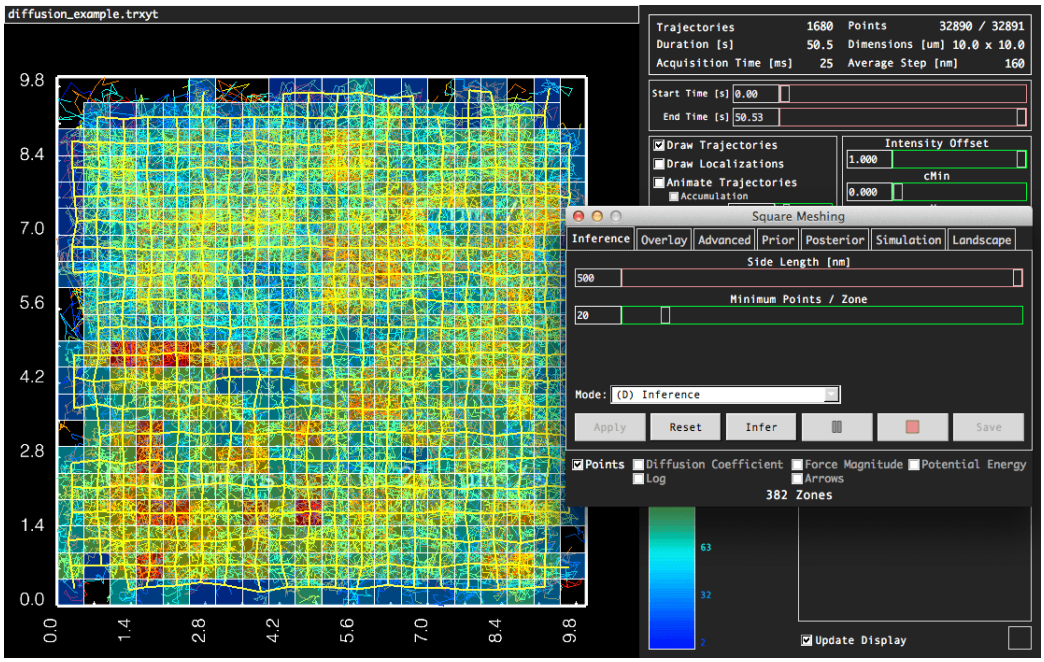
3. Custom Selection Inference

The diffusion coefficient for a custom-selected zone can be estimated using the **Freehand Panel**. As the localization precision is already specified to 30 nm in the file, there is no need to adjust the **Localization Precision [nm]** slider. Select the **Make Selection** button, and draw a region inside the top left quadrant of the trajectory file by pressing and holding the left mouse button. Release the left mouse button to close the region. Now press the **Infer (D,F)** button to infer the diffusion and force inside the selected region. A diffusion value very close to the specified 1.0 $\mu m^2/s$ should be estimated (displayed in the right side of the **Freehand Panel**. Notice that the directional bias (indicated by the yellow arrow) is extremely weak in magnitude (indicated in the **Freehand Selection** panel, not by the size of the arrow!). This calculation can be redone for the different quadrants, giving accurate estimates of the specified diffusivity.



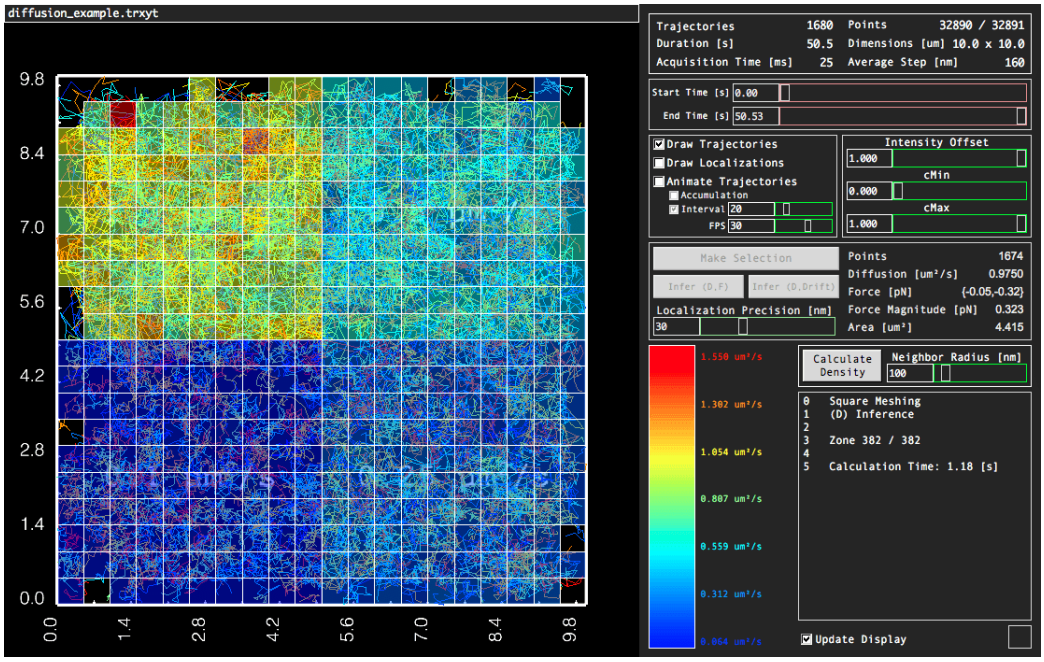
4. Generate a Square Mesh

Deselect the **Make Selection** button in the **Main Interface**, and open the **Square Meshing** interface by accessing **File** > **Inference** > **Meshing** > **Square**. As we know the diffusivity to be relatively constant within each quadrant, we can choose relatively large grid spacings in the mesh. In the **Side Length [nm]** slider, manually type in 500 nm. Afterwards, press the **Apply** button to generate the mesh. The default colorcode corresponds to the number of points in each zone (seen in the **Colorbar**).



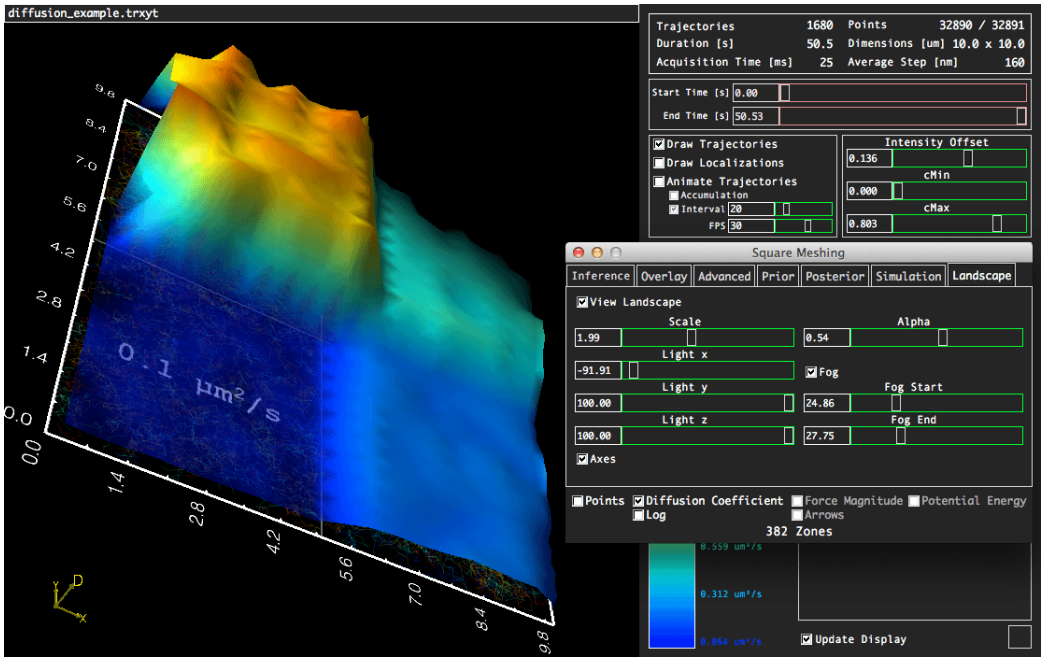
5. Perform Inference

Make sure that the **(D) Inference** is selected in the **Mode** drop down menu in the meshing interface. The localization precision may be adjusted in the **Advanced Tab** of the meshing interface, however, its default value is set to 30 nm which corresponds to the value in the trajectory file. Press the **Infer** button to infer the diffusion coefficient in each of the zones in the mesh. The **Colorbar** will automatically update to the diffusion values. Notice that values in each of the zones accurately corresponds to the diffusion specified in the file.



6. Landscape Viewing

For more intuitive visualization, the diffusion map can be viewed in a 3D landscape mode. Select the **Landscape** tab in the meshing interface, and press **View Landscape**. The **Display Window** becomes manipulatable in 3D, clearly showing the differences in diffusion between the different quadrants. Sliders in the **Landscape** tab can be used to adjust some of the landscape display properties.

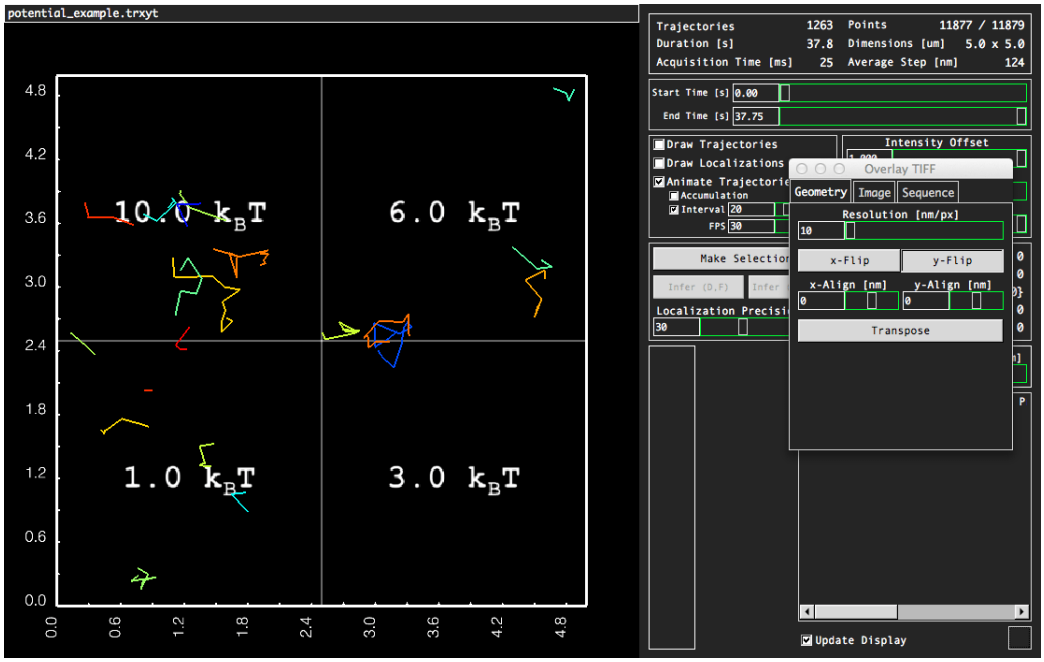


15.2 Example 2: Generating a Potential (Interaction) Energy Map

This example demonstrates how to generate a potential (interaction) energy map of simulated molecule trajectories with realistic diffusivities. The trajectory file is partitioned into four quadrants, each containing a potential energy well with a different depth (the specified depth value is indicated in the background image). The form of each well is Gaussian, with a variance of 300 nm. The localization precision for the trajectories is specified to be 30 nm. The trajectories are set to diffuse at $0.2\mu m^2/s$.

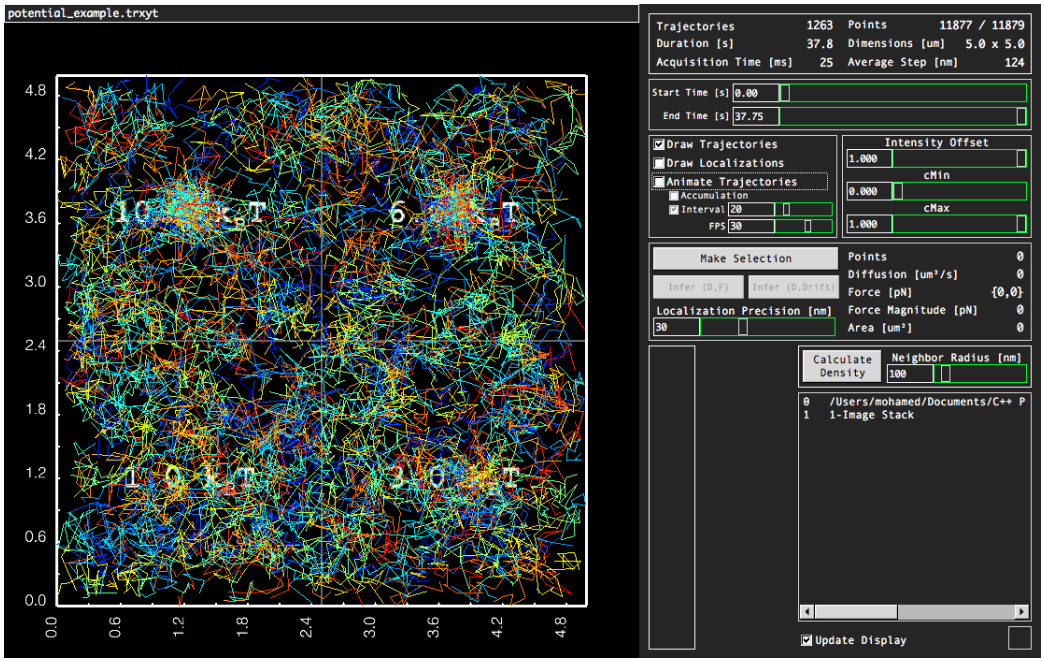
1. Load Potential Example File

Select the **Potential Example (Simulation)** file from the **File** > **Examples** menu. The trajectories will start animating. The specified potential energy depths for the respective quadrants is overlaid as a background image.



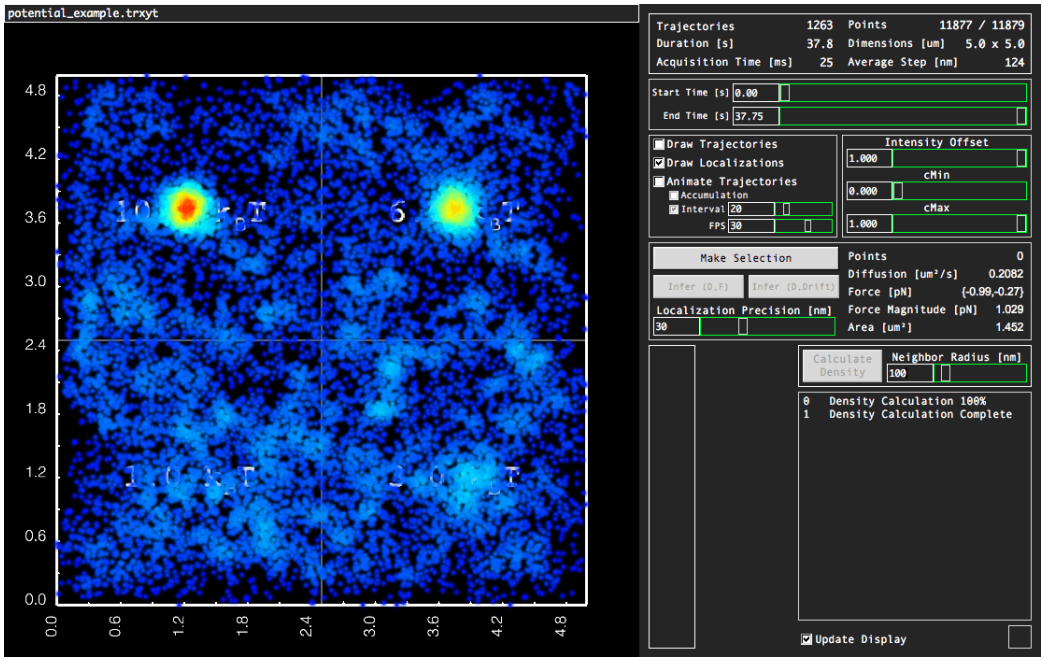
2. Visualize Trajectories

Deselect the **Animate Trajectories** button, and select the **Draw Trajectories** button in the **Main Interface** to overlay all the trajectories in the file.



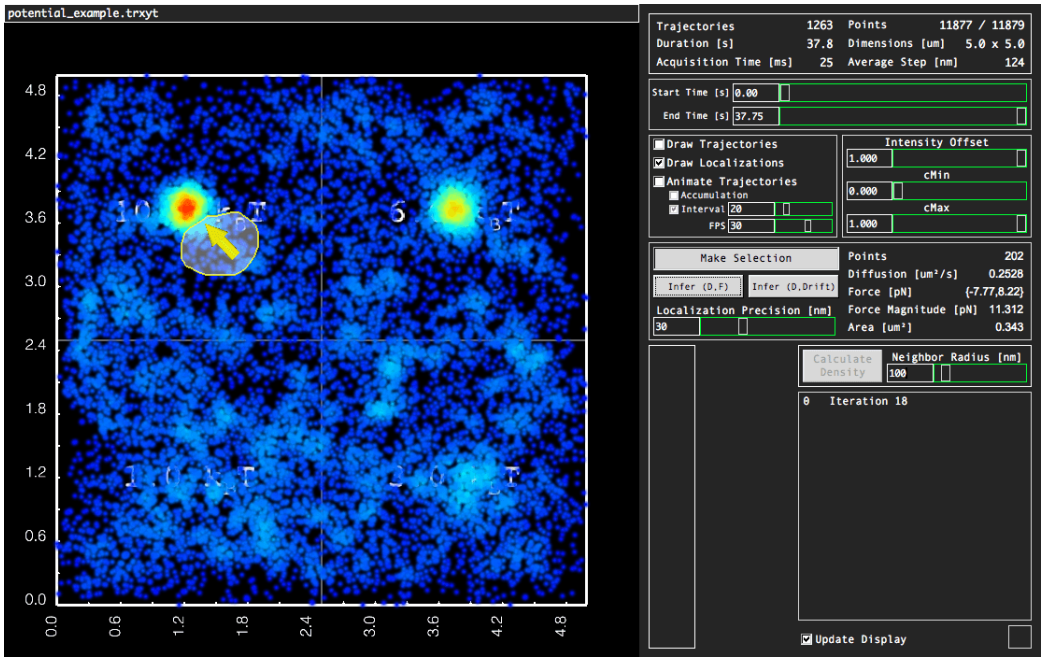
3. Calculate Density

To help reveal regions of attraction, we can perform a naive density calculation to show where trajectory points are most concentrated. Press the **Calculate Density** button in the **Density Panel** of the **Main Interface** (here we leave the **Neighborhood Radius** to its default value. Upon calculation, the colorcode will correspond to the relative density of each localization in the loaded trajectory file, red being the highest and dark blue being the lowest.



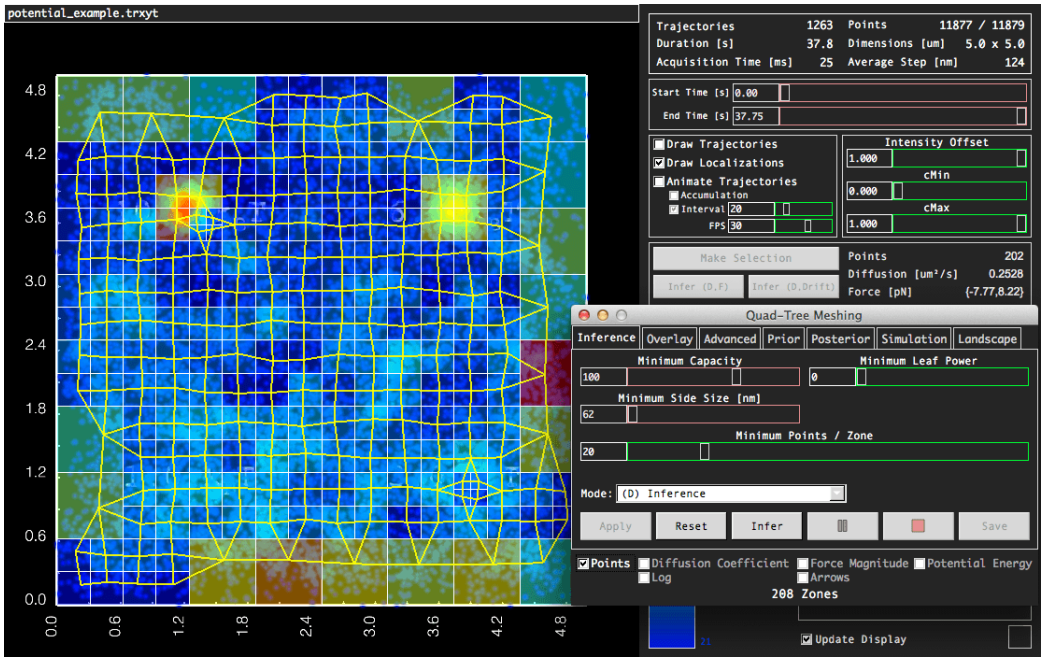
3. Custom Selection Inference

Before generate the potential energy map, we can infer directional bias of trajectories at the proximity of high density localization regions. The force for a custom-selected zone can be estimated using the **Freehand Panel**. As the localization precision is already specified to 30 nm in the file, there is no need to adjust the **Localization Precision [nm]** slider. Select the **Make Selection** button, and draw a region adjacent to a high density region revealed by the density calculation. Now press the **Infer (D,F)** button to infer the diffusion and force inside the selected region. For the deepest wells, the force is strong in magnitude and strongly directional towards the well (displayed in the right side of the **Freehand Panel**. The more shallow wells will show weaker tendencies.



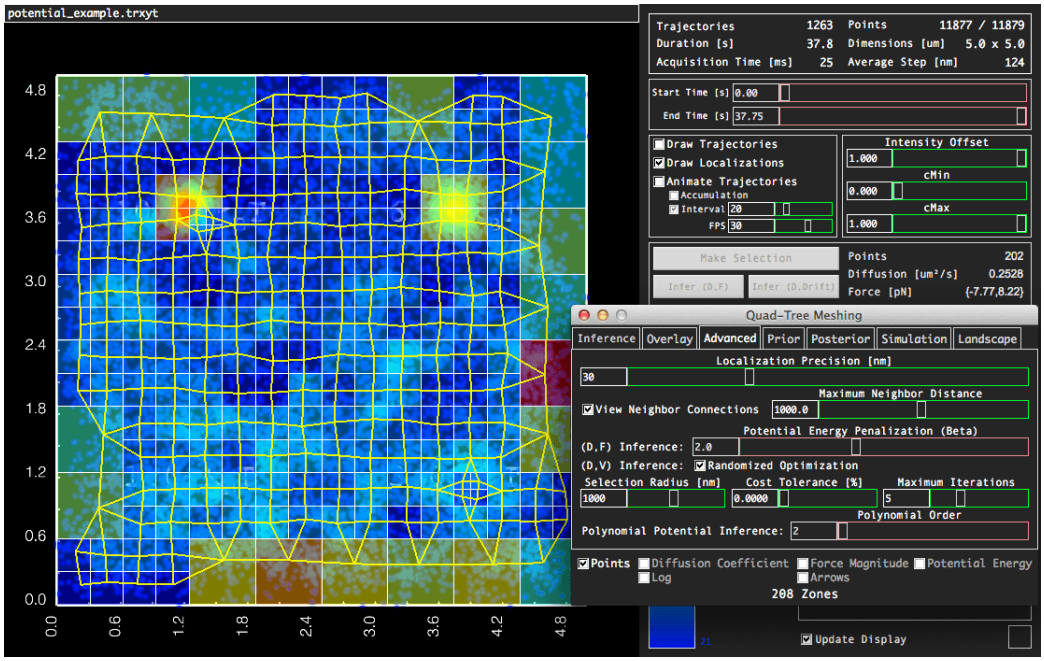
5. Generate a Quad-Tree Mesh

Deselect the **Make Selection** button in the **Main Interface**, and open the **Quad-Tree Meshing** interface by accessing **File** > **Inference** > **Meshing** > **Quad-Tree**. Here we opt for the default meshing values, although the user is encouraged to test different values. Press the **Apply** button to generate the mesh. The yellow lines connecting each of the zones corresponds to which neighboring zones "see each other" (Described in Section 6.4.1). The default colorcode corresponds to the number of points in each zone (seen in the **Colorbar**).



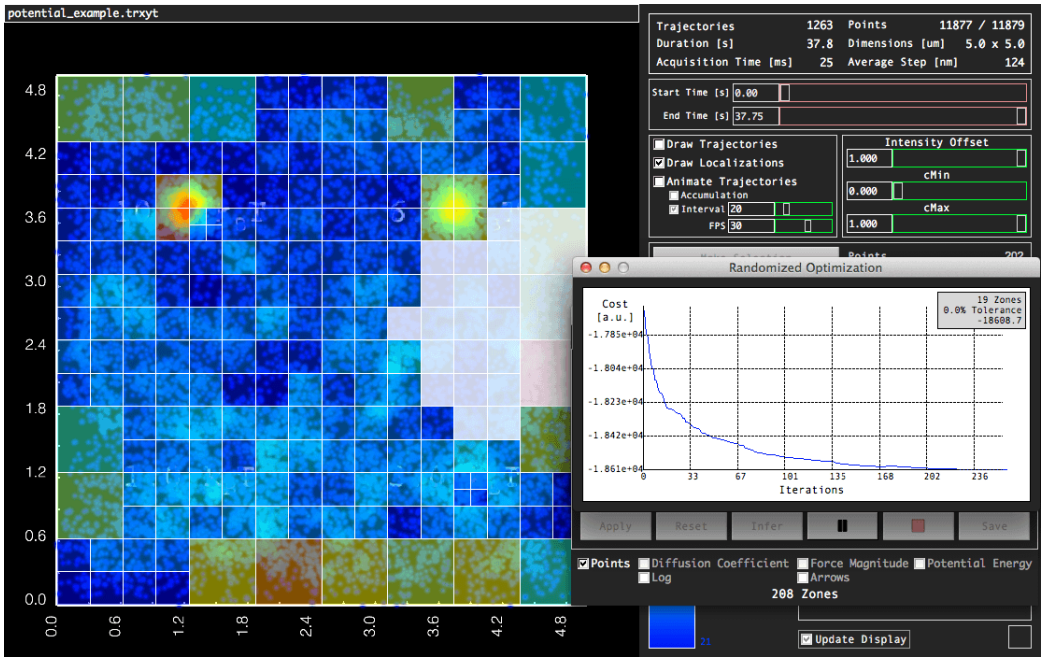
6. Setup a Randomized Optimization Calculation

Select the **(D,V) Inference** option from the **Mode** drop down menu in the meshing interface. The localization precision may be adjusted in the **Advanced Tab** of the meshing interface, however, its default value is set to 30 nm which corresponds to the value in the trajectory file. As the **(D,V) Inference** calculation is quite computationally expensive, we choose to perform a randomized optimization which greatly reduces the time needed to infer parameters from the trajectories. In the **Advanced** tab, select the **Randomized Optimization** button, and increase the **Selection Radius [nm]** slider to roughly 1000 nm and reduce the **Cost Tolerance [%]** slider to zero (implying we will have to manually stop the calculation).



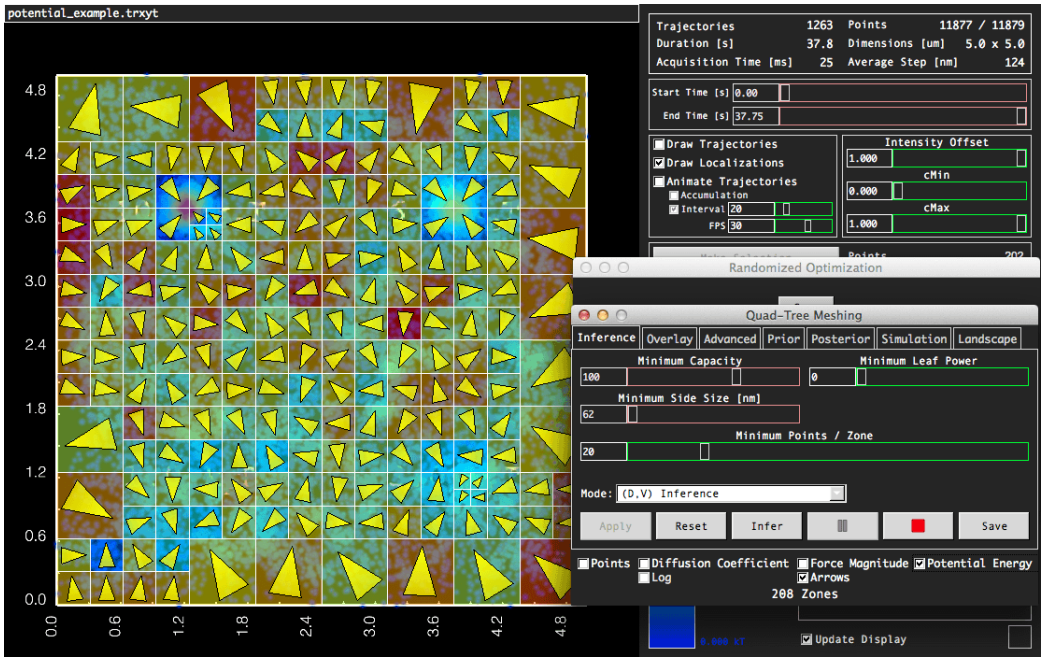
7. Perform Inference

Select the **Inference** tab and press the **Infer** button to infer the diffusion, forces, and potential energy in each of the zones in the mesh using randomized optimization. Subregions of the mesh appearing white will appear white indicating that they are being optimized. The idea is to perform the randomized optimization until the cost function has sufficiently decayed. In this example, this corresponds to approximately 200 iterations.



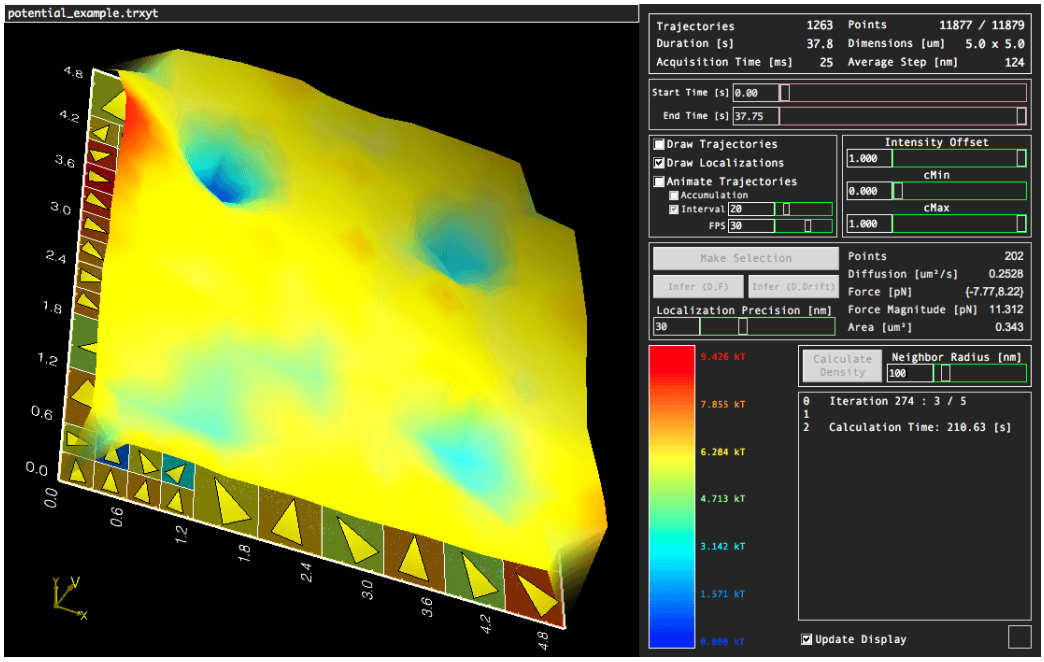
8. Stop Calculation

At roughly 200 iterations in the randomized optimization (indicated in the **Randomized Optimization** window), press the **Pause** (double verticle lines) button then the **Stop** (red square) to stop the inference calculation. Press the **Potential Energy** button to overlay the potential energy values to the mesh, with yellow arrows corresponding to the force direction overlaid. Clearly seen from the yellow force arrows is the directional bias of the trajectories. In fact, for the deep wells will have a strong bias, whereas the weaker ones will not. In the bottom left quadrant, the depth of the potential well corresponds to 1.0 k_BT which is roughly the background potential energy due to thermal fluctuations.



9. Landscape Viewing

For more intuitive visualization, the potential energy map can be viewed in a 3D landscape mode. Select the **Landscape** tab in the meshing interface, and press **View Landscape**. The **Display Window** becomes manipulatable in 3D, clearly showing the different potential energy wells between the different quadrants. Sliders in the **Landscape** tab can be used to adjust some of the landscape display properties. Here, we see that the approximate depths of the potential energy wells correspond to the specified values.

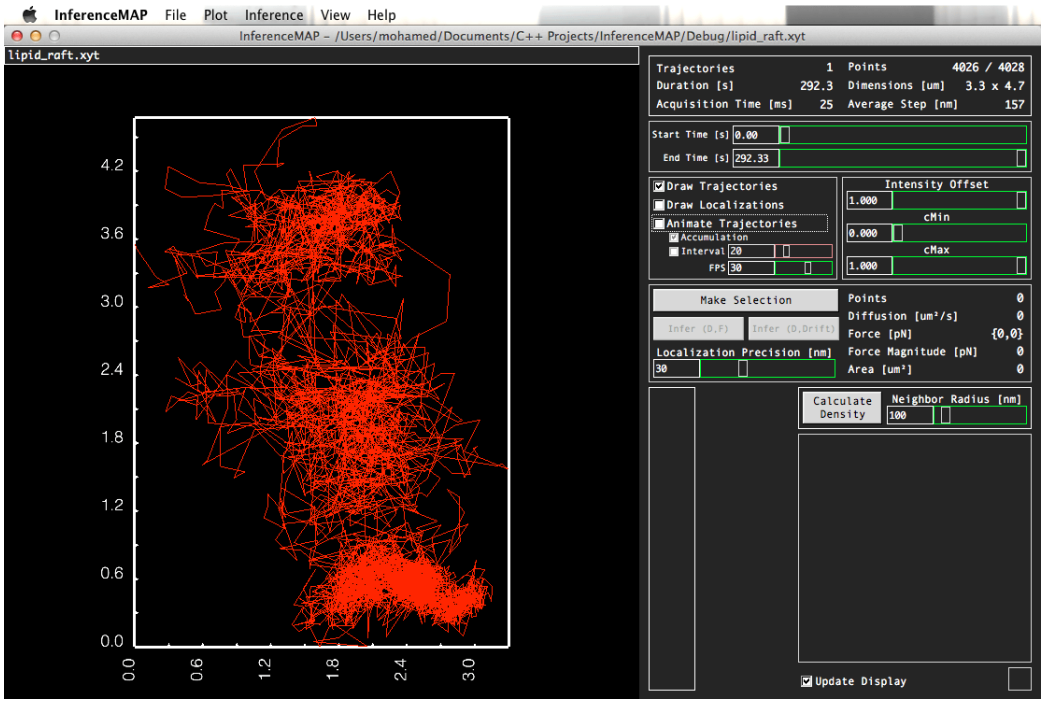


15.3 Example 3: Membrane Microdomain

This example demonstrates how to generate a potential (interaction) energy map based on the trajectory of an ϵ-toxin receptor tagged with an amine-coated lanthanide oxide nanoparticle on an MDCK cell [6]. The receptor is seen to hop between three different lipid rafts, each of which having a different strength of confinement. Experimentally, these trajectories were captured in a wide field fluorescent microscope configuration.

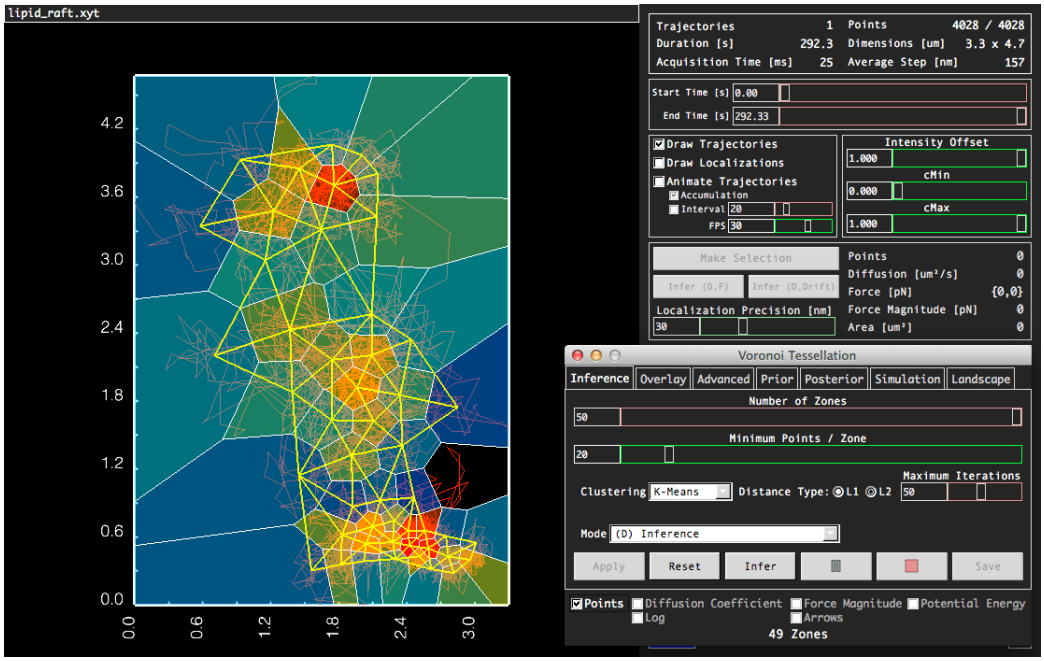
1. Load Toxin Receptor Example File

Select the **Toxin Receptor in Lipid Raft)** file from the **File > Examples** menu.



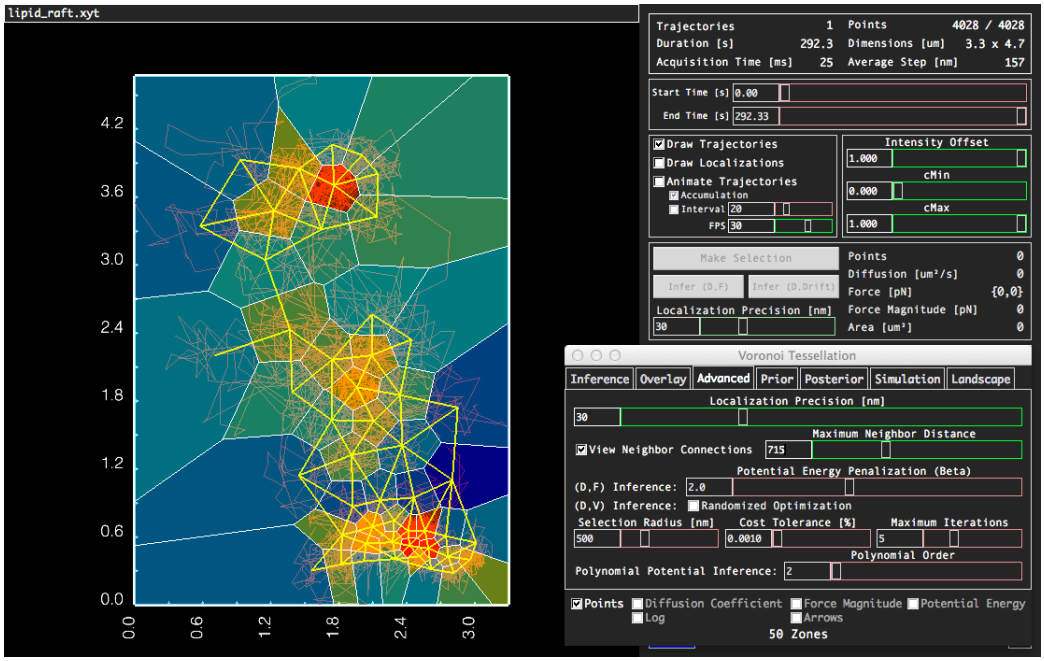
2. Generate a Voronoi Mesh

Open the **Voronoi Meshing** interface from the **File > Inference > Meshing > Voronoi** menu. For this meshing mode, the number of zones must be predefined beforehand. **InferenceMAP** automatically selects an appropriate number based on the number of localizations in the loaded data set. Here we shall use the default value of 50 zones. Press the **Apply** button to generate the Voronoi mesh.



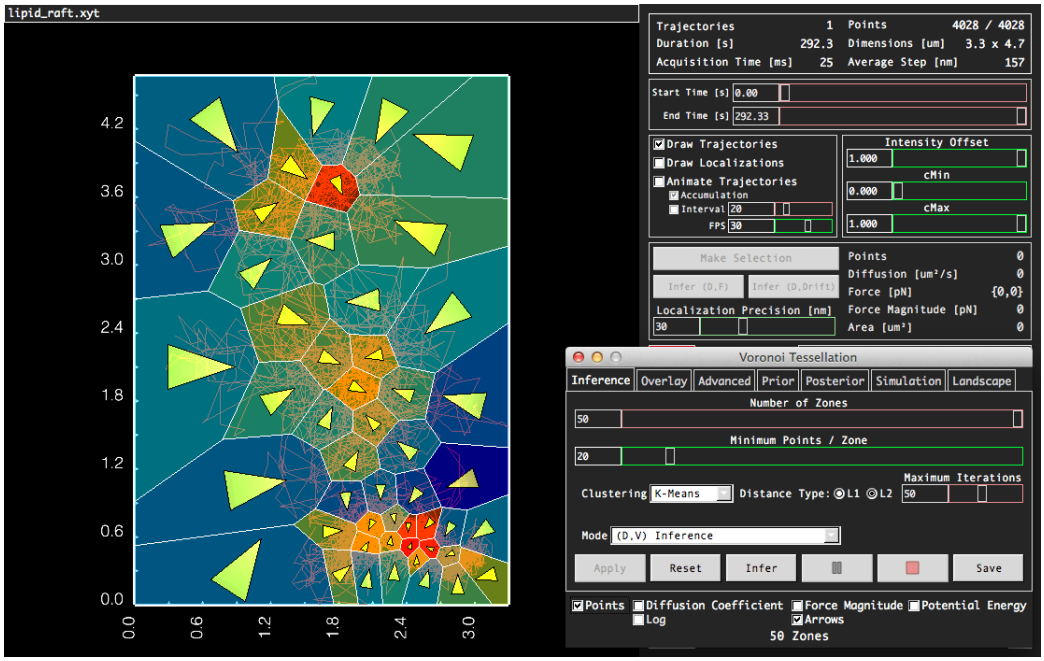
3. Adjust Mesh

The generated mesh is seen to possess a deactivated zone in the bottom right corner. Activate this zone by right clicking on it. The yellow lines connecting each of the zones corresponds to which neighboring zones "see each other" (Described in Section 6.4.1). In this case, some of the connections between zones do not make sense based on the layout of the trajectory (e.g. the top two rafts do not seem to be associated). Select the **Advanced** tab and adjust **Maximum Neighbor Distance [nm]** to roughly 715 nm. Additionally, adjust the **Localization Precision [nm]** slider to 20 nm (corresponding to the approximate experimental value).



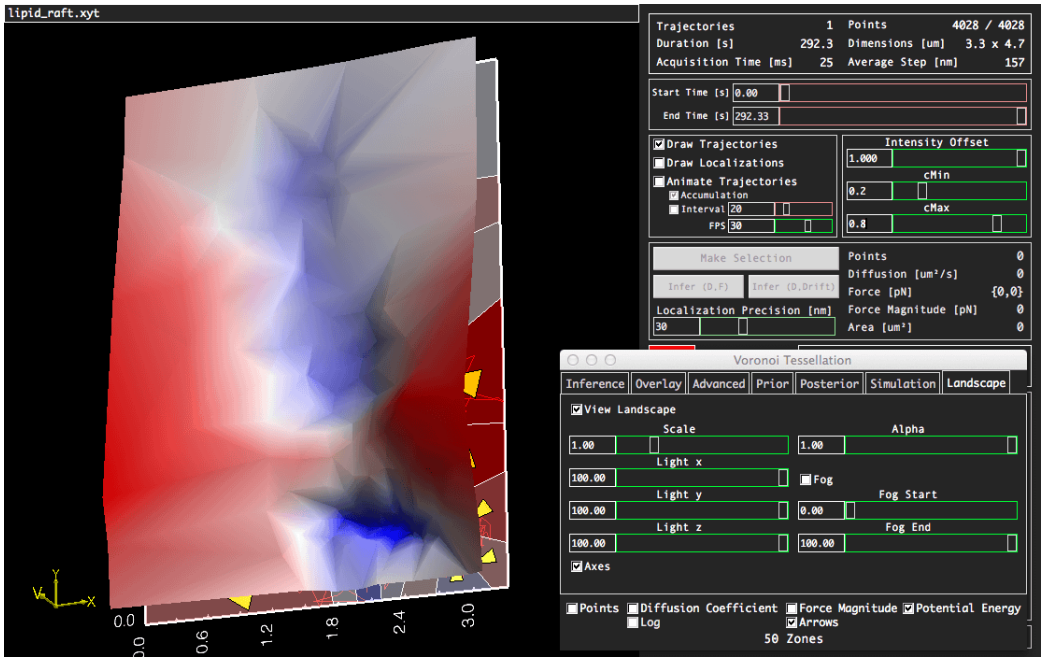
4. Perform Inference

In the **Inference** tab, select the **(D,V) Inference** mode. To perform the inference calculation, press **Infer**. The **(D,V) Inference** mode is generally a slow and expensive calculation and necessitates the randomized optimization feature to complete in a reasonable amount of time. However, as there are only 50 zones in this mesh, the calculation time will not be too lengthy (roughly 2 minutes).



5. Landscape Viewing

Based on the force arrows it is seen that the toxin receptor is separately confined in three distinct parts of the trajectory. To see the potential energy field which gives rise to this confining force, select **Potential Energy** in the bottom of the meshing interface. Adjust the **cMin** and **cMax** sliders in the **Main Interface** to 0.2 and 0.8, respectively. Now select the **Overlay** tab, and choose **Blue Red Map** drop-down menu. Next, select the **Landscape** tab and press **View Landscape**. The **Display Window** becomes manipulatable in 3D. Represented are three potential energy wells, the bottom most one being much deeper (roughly 5 k_BT) than the top two (roughly 2 k_BT). Essentially, the bottom strongly stabilizes the toxin receptor, while the top two wells are too shallow to keep the receptor confined for long.

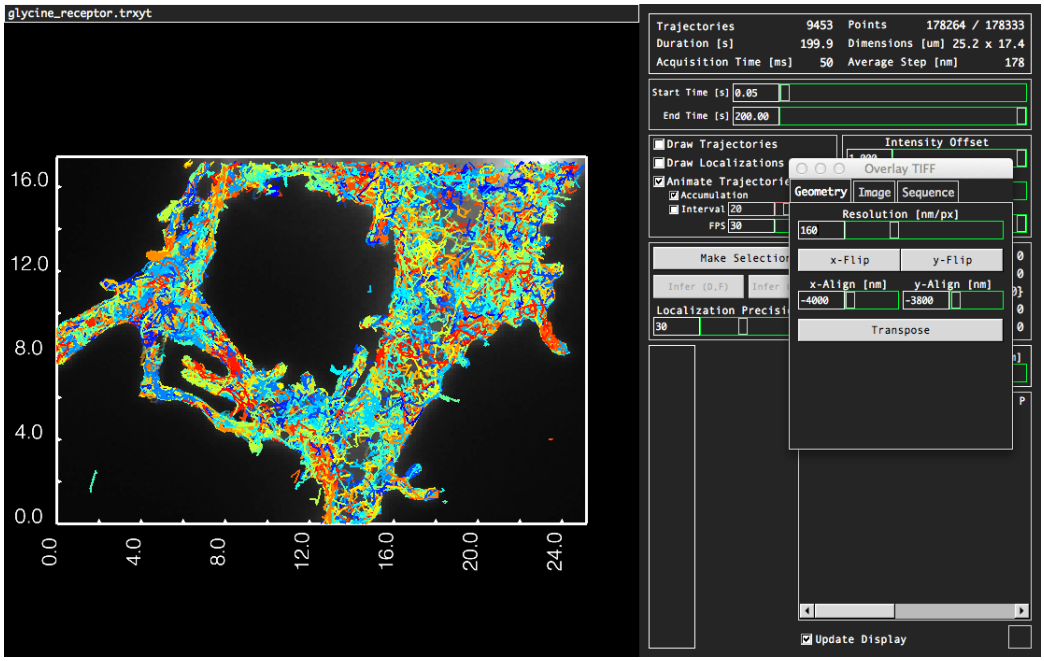


15.4 Example 4: Neurotransmitter Receptors

This example demonstrates how to create a diffusion map for glycine receptors on a mouse hippocampal neuron membrane. Experimentally, these trajectories were captured in a uPAINT configuration, in a similar fashion to [1].

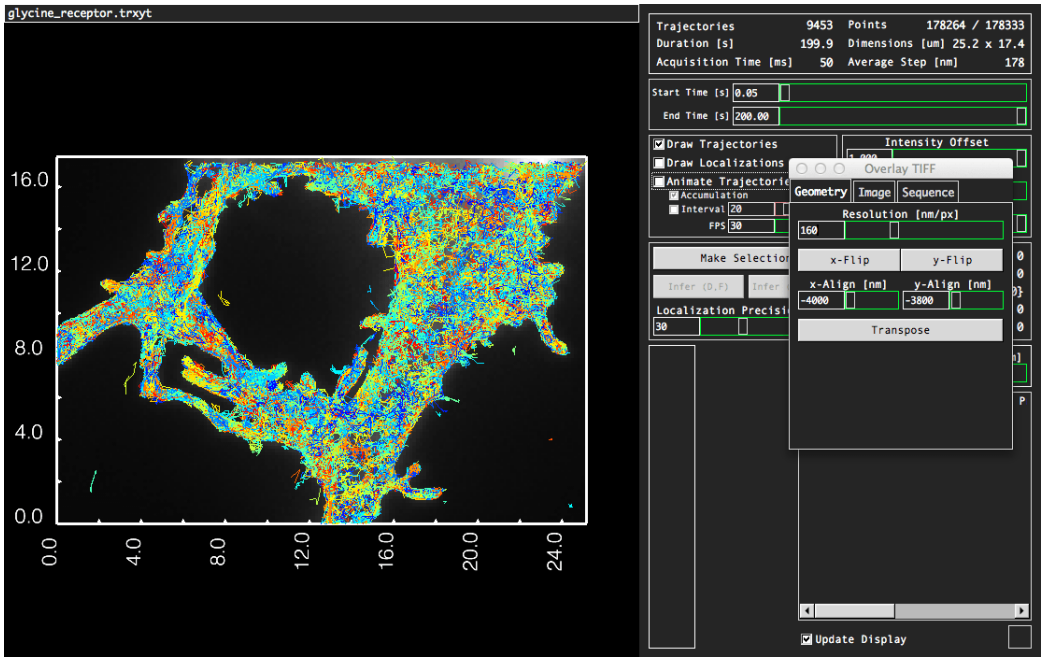
1. Load Neurotransmitter Receptor File

Select the **Glycine Receptors on Mouse Hippocampal Neuron)** file from the **File > Examples** menu. The trajectories will start animating in an accumulation mode, demonstrating how rapidly the surface of the apical membrane is explored by the tracked glycine receptors. A GFP image is overlaid to show the shape of the neuron.



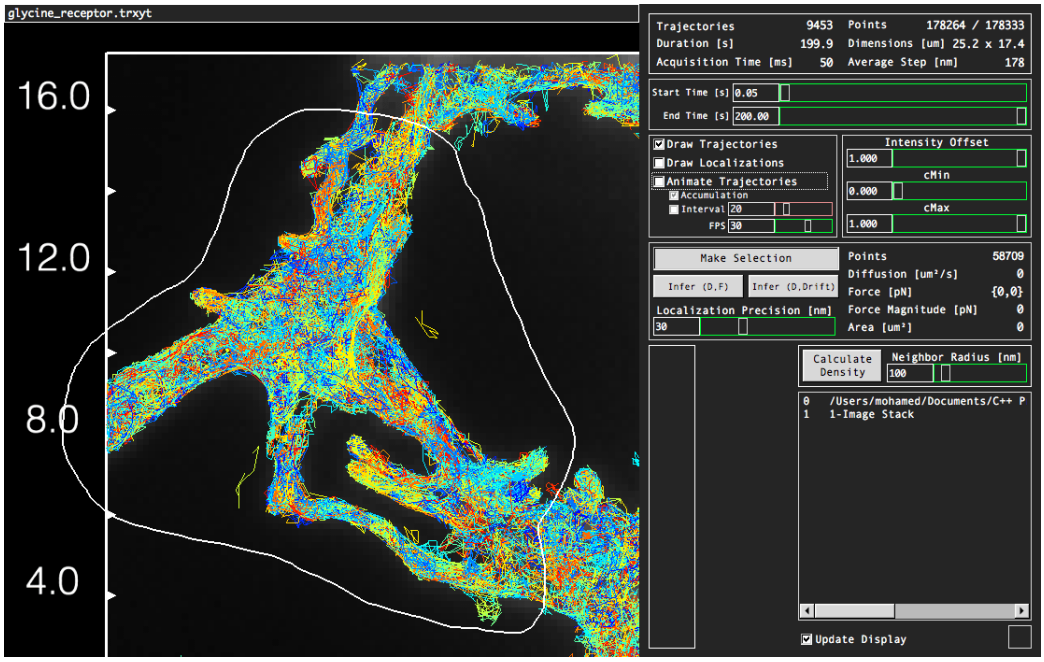
2. Visualize Trajectories

Deselect the **Animate Trajectories** button, and select the **Draw Trajectories** button in the **Main Interface** to overlay all the trajectories in the file.



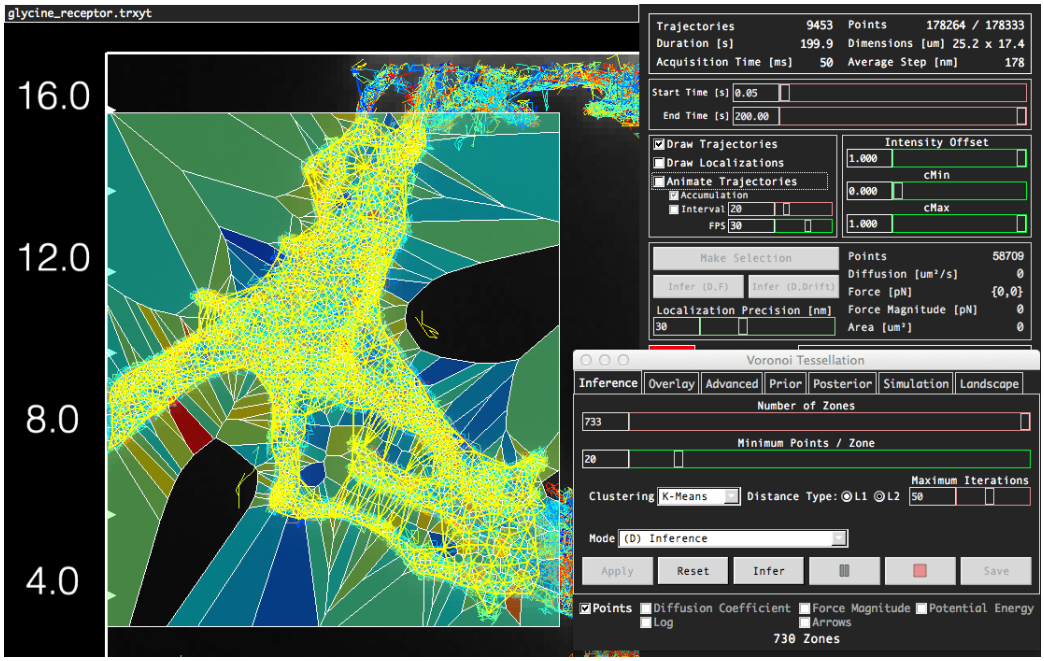
3. Make Custom Selection

Zoom into the region to the left by right clicking and dragging the mouse to generate a box in the **Display Window**. Letting go will zoom into the selected region (double-clicking will reset the view if you wish to re-select). Press the **Make Selection** button in the **Freehand Panel**, and select a region surrounding the part of the neuron indicated in the figure.



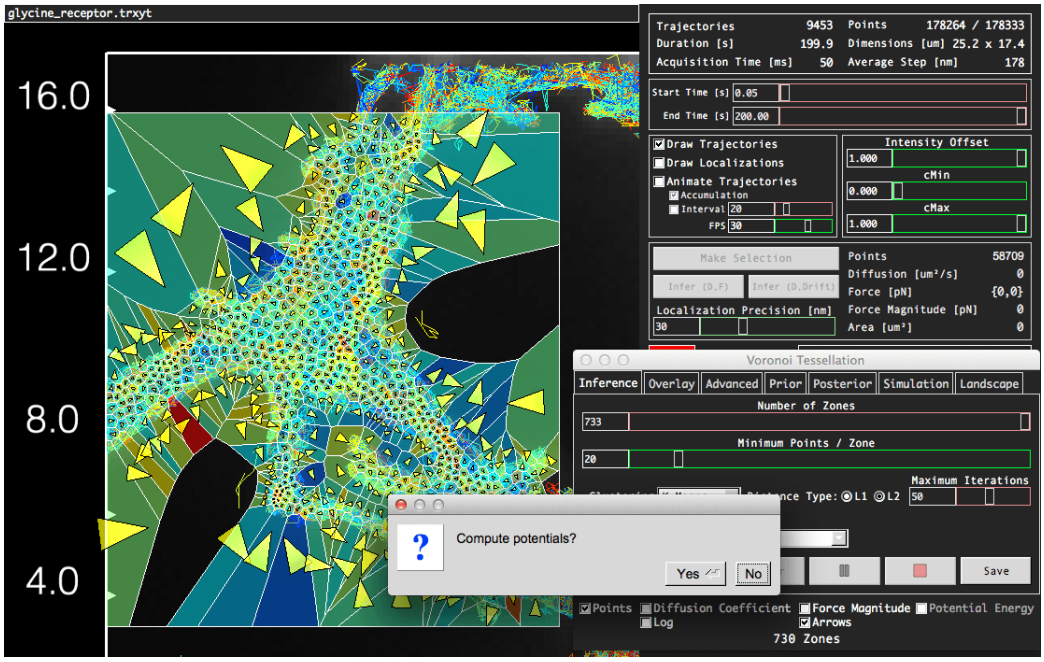
4. Generate a Voronoi Mesh

In this example Voronoi meshing will be used on the user-selected region of the trajectories. Open the **Voronoi Meshing** interface from the **File** > **Inference** > **Meshing** > **Voronoi** menu. For this meshing mode, the number of zones must be predefined beforehand. **InferenceMAP** automatically selects an appropriate number based on the number of localizations in the loaded data set. Here we shall use the default value. Press the **Apply** button to generate the Voronoi mesh in only the custom selected region, which may take a couple minutes.



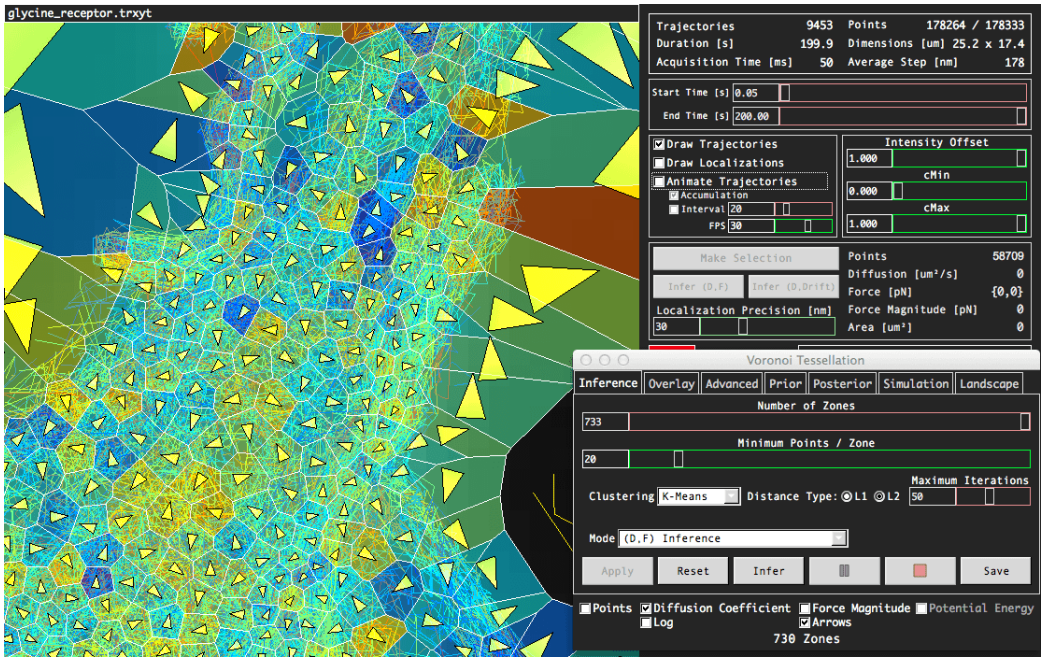
5. Perform Inference

The yellow lines connecting each of the zones corresponds to which neighboring zones "see each other" (Described in Section 6.4.1). However, we shall be using **(D,F) Inference** in this example, which does not require zones to "see each other". Select the **(D,F) Inference** in the mode drop-down menu. Press **Infer** to perform the inference. Select **No** when prompted to compute the potentials.



5. Viewing Biases in Motion

Select the **Diffusion Coefficient** button in the bottom of the meshing interface. Create a box to zoom in on parts of the dendrites of the neuron. Directional arrows indicate regions where motion is systematically biased in a certain directions (animating the trajectories will confirm this).



References

[1] Jean-Baptiste Masson, Patrice Dionne, Charlotte Salvatico, Marianne Renner, Christian G. Specht, Antoine Triller, and Maxime Dahan, "Mapping the Energy and Diffusion Landscapes of Membrane Proteins at the Cell Surface Using High-Density Single-Molecule Imaging and Bayesian Inference: Application to the Multiscale Dynamics of Glycine Receptors in the Neuronal Membrane", *Biophysical Journal* 106 (1), 74–83 (2014).

In this paper we extend the inference scheme to larger scales, up to the full mapping of cell membranes. We show the efficiency of the inference at various scales and for very heterogeneous environments. We demonstrate the sensitivity of the method on experimental data by proving that diffusion and potential maps in synapses can be quantified for various mutants of the glycine receptor. Furthermore, we demonstrate that Gillespie schemes can be used to simulate trajectories in the inferred maps. Numerous simulations in the supplementary materials prove the efficiency of the scheme and discuss possible bias. We also detail more strategies to design priors such as Jeffreys' prior and regularization schemes for the fields.

[2] Maximilian U. Richly, Silvan Türkcan, Antoine Le Gall, Nicolas Fiszman, Jean-Baptiste Masson, Nathalie Westbrook, Karen Perronet, and Antigoni Alexandrou, "Calibrating optical tweezers with Bayesian Inference", *Optics Express* 21 (25), 31578–31590 (2013).

In this paper we show that optical tweezers (confinement and probing done with the same laser) can be calibrated more precisely than previously used method using a Bayesian inference scheme.

[3] Silvan Türkcan and Jean-Baptiste Masson, "Bayesian Decision Tree for the Classification of the Mode of Motion in Single-Molecule Trajectories", *PLoS ONE* 8 (12), e82799 (2013).

In this paper we use the inference and various statistical estimators to quantify the nature of biomolecule motion from freely diffusing to harmonically confined. We also apply it to time changing states.

[4] Silvan Türkcan, Maximilian U. Richly, Antigoni Alexandrou, and Jean-Baptiste Masson, "Probing Membrane Protein Interactions with Their Lipid Raft Environment Using Single-Molecule Tracking and Bayesian Inference Analysis", *PLoS One* 8 (1), e53073 (2013).

In this paper we analyse the hopping of biomolecules between confining environments. We demonstrate that the amount of energy necessary to hop can be accurately inferred, that the diffusivity map can be accurately inferred even in the hopping area and that the inference can be applied with time varying diffusivity and potential fields. The method is applied to the hopping of ϵ -toxin. Time evolving fields are induced by the removal of cholesterol and sphingolipids by respectively cholesterol oxidase and sphingomyelinase. Supplementary materials give a large amount of simulations supporting our claims.

[5] Silvan Türkcan, Antigoni Alexandrou, and Jean-Baptiste Masson, "A Bayesian Inference Scheme to Extract Diffusivity and Potential Fields from Confined Single-Molecule Trajectories", *Biophysical Journal* 102 (10), 2288–2298 (2012).

We give more details on the implementation of the inference method, the effect of positioning noise, the effect of confinement, the correction function for highly confined motion, and the projection of the confining potential on polynomials to increase precision of the confining potential. Furthermore, it is demonstrated that the inference does not generate potentials or force fields if there are no fields interacting with the random walker.

[6] Silvan Türkcan, Jean-Baptiste Masson, Didier Casanova, Geneviève Mialon, Thierry Gacoin, Jean-Pierre Boilot, Michel R. Popoff, Antigoni Alexandrou, "Observing the Confinement Potential of Bacterial Pore-Forming Toxin Receptors Inside Rafts with Nonblinking Eu^{3+}-Doped Oxide Nanoparticles", *Biophysical Journal* 102 (10), 2299–2308 (2012).

In this paper we apply the method on the ϵ-toxin receptor motion inside a lipid raft, showing the diffusivity map in the raft and the non-local confining potential in the raft.

[7] Guillaume Voisinne, Antigoni Alexandrou, and Jean-Baptiste Masson, "Quantifying Biomolecule Diffusivity Using an Optimal Bayesian Method", *Biophysical Journal* 98 (4), 596–605 (2010).

In this paper we demonstrate the optimality of the Bayesian inference method to infer diffusivity. Furthermore, we quantify the acquisition of information gathered via inference (Fisher Information) and give analytical solutions for various confinement geometries.

[8] Jean-Baptiste Masson, Didier Casanova, Silvan Türkcan, Guillaume Voisinne, Michel R. Popoff, Massimo Vergassola, and Antigoni Alexandrou, "Inferring Maps of Forces Inside Cell Membrane Microdomains", *Physical Review Letters* 102 (4), 048103 (2009).

The original paper in which the Bayesian Inference method for single molecule trajectory, as well as the Langevin modelling and the solution of the Fokker-Planck equation, are introduced. We prove that both diffusivity and force fields can be inferred accurately and the scheme is applied to test trajectories of the receptor of the ϵ-toxin.

16 Acknowledgments

16.1 Development Libraries

Various freely-available development libraries were used in **InferenceMAP** . They are referenced below. In all cases, **InferenceMAP** conforms to all license agreements, and no modifications to the original authors' libraries are made.

16.2 FLTK

The Fast Light Toolkit (**FLTK**) is the cross-platform graphical user interface library used in **InferenceMAP** . It was originally developed by Bill Spitzak.

16.3 OpenGL

OpenGL is the graphics library used for displaying graphics in **InferenceMAP** . It is maintained by the **Khronos Group**.

16.4 Libtiff

Libtiff is the library used for writing Tagged Image File Format files using **InferenceMAP** . It was originally developed by Sam Leffler at Silicon Graphics.

16.5 TexFont

TexFont is the textured font library used in **InferenceMAP** to display text in the display window. It was originally developed by Mark Kilgard.

16.6 Clustering Algorithms

Clustering algorithms (K-Means and H-Means) available for the voronoi tessellation modes are based on the C++ code by John Burkardt (original Fortran Code by David Sparks), available at: `http://people.sc.fsu.edu/~jburkardt/c_src/asa136/asa136.html`.

17 Contact Information

Mohamed El Beheiry (Developer) mohamed.elbeheiry@gmail.com

Jean-Baptiste Masson jbmasson@pasteur.fr